

# Universidad de Alcalá

## Escuela Politécnica Superior

### Máster Universitario en Ingeniería de Telecomunicación

#### Trabajo Fin de Máster

Diseño de una arquitectura para un enlace PLC de banda ancha  
de alta velocidad para entornos vehiculares o aeroespaciales

**Autor:** Francisco Membibre Fernández

**Tutor:** Álvaro Hernández Alonso

2019



# UNIVERSIDAD DE ALCALÁ

## ESCUELA POLITÉCNICA SUPERIOR

**Máster Universitario en Ingeniería de Telecomunicación**

**Trabajo Fin de Máster**

**Diseño de una arquitectura para un enlace PLC de banda ancha  
de alta velocidad para entornos vehiculares o aeroespaciales**

Autor: Francisco Membibre Fernández

Director: Álvaro Hernández Alonso

**Tribunal:**

**Presidente:** Raúl Mateos Gil

**Vocal 1º:** Manuel Prieto Mateo

**Vocal 2º:** Álvaro Hernández Alonso

Calificación: .....

Fecha: .....





# Resumen

Este trabajo tiene como finalidad realizar comunicaciones PLC de banda ancha y alta velocidad en entornos vehiculares o aeroespaciales. Para ello, se utiliza la técnica de acceso al medio FBMC recomendada por el estándar IEEE 1901-2010 para comunicaciones PLC. Para la etapa de conversión se utilizan dos Analog Front-End que constan de conversores AD/DA, pero un único bus de datos bidireccional. Es por tanto necesario realizar una arquitectura que conste de dos drivers, que configuren un AFE como transmisor y otro como receptor; una interfaz para interactuar entre todos los módulos del diseño y un módulo de Sincronismo para determinar el inicio de la comunicación entre el transmisor y el receptor del transmultiplexor FBMC. La arquitectura se realizará en una FPGA y será integrada en un SoC para su gestión avanzada.

**Palabras clave:** Power Line Communications (PLC), Analog Front-End (AFE), System-on-Chip (SoC), Field-Programmable Gate Array (FPGA), Filter Bank Multi-carrier (FBMC).



# Abstract

The purpose of this work is to carry out high-speed broadband PLC communications in vehicular or aerospace environments. For this purpose, FBMC is used as medium access technique, as recommended by the IEEE 1901 standard for PLC communications. For the conversion stage, two Analog Front-End are used, consisting of AD/DA converters, but a single bidirectional data bus. It is therefore necessary to make an architecture consisting of two drivers, configuring one AFE as a transmitter and another as a receiver, an interface to interact between all modules of the design and a Synchronism module to determine the start of communication between the transmitter and receiver of the transmultiplexor FBMC. The architecture will be implemented in an FPGA and integrated in a SoC for its advanced management.

**Keywords:** PLC, AFE, SoC, FPGA, FBMC.



# Resumen extendido

Las comunicaciones PLC son una importante línea de trabajo e investigación, debido a que utilizan la infraestructura existente de la red eléctrica. Las aeronaves, como muchos otros dispositivos, requieren de alimentación para su funcionamiento, por tanto este canal se encuentra presente y se pueden realizar comunicaciones de banda ancha y alta velocidad por él. Dicho canal presenta una gran cantidad de problemas, debido a que no es un medio creado originalmente para la transmisión de información, además del ruido presente en el mismo, como suma de varios tipos. Para solventar dichos problemas se utilizarán Multi-Carrier Modulation (MCM), en concreto se empleará FBMC, siendo una de las técnicas de acceso al medio recomendada por el estándar de comunicaciones PLC de banda ancha IEEE 1901-2010, como evolución de Orthogonal Frequency Division Multiplexing (OFDM). La utilización de estas modulaciones multiportadora supone una carga computacional muy elevada, por lo que una importante alternativa para su implementación pueden ser los Systems-on-Chip (SoC). Estos dispositivos incorporan una parte lógica, donde se integra la arquitectura que soporta la mayor carga de trabajo de la comunicación; y una software, donde se realiza la gestión de dicha arquitectura programando uno de los procesadores que incorpora. Como elemento intermedio entre la arquitectura y la red eléctrica, se utilizarán dos Analog Front-End, uno para transmitir y otro para recibir al/del canal PLC. Estos elementos incorporan en un mismo dispositivo las etapas de conversión, integración y amplificación que se requieren para abordar estas comunicaciones PLC, por lo que simplifican este apartado. El AFE utilizado es el MAX2981 de Maxim Integrated, que incorpora ambos conversores DA/AD al poder funcionar en cualquier sentido de la comunicación. Para poder establecer el AFE como transmisor o receptor, es necesario fijar un valor determinado para las señales que se aplican a los terminales que componen el chip. Por ello, se requiere introducir dos drivers en la arquitectura que realicen esta tarea al plantearse en este trabajo un sistema de comunicación completo (transmisor y receptor). Estos dispositivos cuentan con registros internos con los cuales se pueden cambiar las características de funcionamiento del mismo; los drivers incluirán una arquitectura que implementa un protocolo SPI para transmitir el valor de los registros y obtener así la nueva funcionalidad. Entre el transmultiplexor FBMC y los AFE, se encuentra la arquitectura que plantea este trabajo, que está compuesta por un bloque transmisor y otro receptor. El bloque transmisor incluye el propio transmisor FBMC, una

interfaz y un driver para establecer uno de los AFE como transmisor. El bloque receptor está compuesto por un driver que establece el AFE restante como receptor, una interfaz, un módulo de sincronismo y el receptor FBMC. La interfaz está desglosada en dos partes, una dentro del bloque transmisor y otra del bloque receptor. En la parte de la interfaz dedicada al transmisor, se tratan las tramas de 512 subportadoras que recibe del transmisor FBMC. Almacenará también dos tramas de sincronismo como preámbulo inicial que se transmitirán en primera instancia, para poder en recepción, detectarlas en el módulo de sincronismo y determinar el inicio de la comunicación. El último módulo de este bloque es un driver, que configura un AFE como transmisor para poder enviar al canal PLC todas las tramas, tanto de sincronismo como de datos. En recepción existe otro driver, pero para configurar el otro AFE como receptor, y poder así recibir las tramas desde el canal PLC. Una vez se encuentran las tramas en el bloque receptor, se almacenan en paralelo en dos memorias, una que actúa como buffer y otra para transmitir bloques de muestras al módulo de sincronismo. Este módulo, se basa en la realización de la correlación cruzada de las tramas de sincronismo recibidas del canal PLC, con las tramas ideales almacenadas en su interior. Si se introduce una trama de sincronismo completa en este módulo, provocará en el dominio del tiempo un pico de correlación, coincidiendo la posición del pico con la última muestra de la trama de sincronismo. Gracias a ello, se determina la posición de la primera subportadora de la primera trama.

# Índice general

Resumen	v
Abstract	vii
Resumen extendido	ix
Índice general	xi
Índice de figuras	xv
Índice de tablas	xxi
Lista de acrónimos	xxv
<b>1 Introducción</b>	<b>3</b>
1.1 Finalidad del proyecto . . . . .	3
1.2 Contexto . . . . .	4
1.3 Entornos de desarrollo . . . . .	7
1.4 Aplicaciones aeroespaciales o vehiculares . . . . .	8
1.5 Estructura del documento . . . . .	10
<b>2 Antecedentes</b>	<b>13</b>
2.1 Comunicaciones PLC (Power Line Communication) . . . . .	13
2.1.1 Regulación de las comunicaciones PLC . . . . .	14
2.1.1.1 Regulación de las comunicaciones PLC de banda estrecha	14
2.1.1.2 Regulación de las comunicaciones PLC de banda ancha . .	15
2.1.2 Comunicaciones PLC de banda estrecha . . . . .	15
2.1.2.1 Estándares para comunicaciones PLC de banda estrecha .	16
2.1.3 Comunicaciones PLC de banda ancha . . . . .	16

2.1.3.1	Estándares para comunicaciones PLC de banda ancha . .	18
2.2	Sistemas SoC (System-On-Chip) . . . . .	19
2.3	Sistemas SoC en comunicaciones PLC . . . . .	21
<b>3</b>	<b>Diseño Propuesto</b>	<b>23</b>
3.1	Introducción . . . . .	23
3.2	Descripción general del diseño . . . . .	25
3.3	Descripción y configuración del diseño en el sentido de la comunicación . .	27
<b>4</b>	<b>Descripción del módulo AFE (Analog Front-End)</b>	<b>33</b>
4.1	Introducción . . . . .	33
4.2	Conexión eléctrica del AFE . . . . .	35
4.2.1	Conectores FMC: HPC y LPC . . . . .	35
4.2.2	Restricciones temporales . . . . .	36
4.3	Registros internos del AFE . . . . .	38
<b>5</b>	<b>AFE Driver</b>	<b>43</b>
5.1	Introducción . . . . .	43
5.2	Configuración del AFE . . . . .	46
5.2.1	Descripción de la arquitectura . . . . .	46
5.2.2	Configuración de la arquitectura vía software . . . . .	49
5.2.3	Cambio de dominio de reloj . . . . .	51
5.3	SPI (Serial Peripheral Interface) . . . . .	53
5.3.1	Descripción de la arquitectura . . . . .	53
5.3.2	Configuración vía software . . . . .	54
5.3.3	Resultados experimentales parciales . . . . .	55
<b>6</b>	<b>Sincronismo</b>	<b>61</b>
6.1	Introducción . . . . .	61
6.2	Estudio teórico . . . . .	62
6.2.1	Tramas de sincronismo . . . . .	62
6.2.2	Método de correlación cruzada . . . . .	62
6.2.3	Transformada Rápida de Fourier . . . . .	63
6.2.4	Celdas DSP48E1 en las arquitecturas de los dispositivos FPGA . .	65



6.3	Modelo en coma fija . . . . .	65
6.3.1	Cuantificación de las tramas de sincronismo transmitidas . . . . .	66
6.3.2	Cuantificación de las tramas recibidas en el bloque receptor y su FFT . . . . .	67
6.3.3	Cuantificación del conjugado de la FFT de las tramas de sincronismo ideales . . . . .	69
6.3.4	Cuantificación de la correlación . . . . .	70
6.3.5	Cuantificación de la IFFT de la correlación . . . . .	72
6.4	Definición de la arquitectura del sincronismo . . . . .	73
6.5	Pruebas experimentales . . . . .	75
<b>7</b>	<b>Módulo de Interfaz</b>	<b>77</b>
7.1	Introducción . . . . .	77
7.2	Interfaz Transmisor . . . . .	79
7.3	Interfaz Receptor . . . . .	81
<b>8</b>	<b>Resultados</b>	<b>87</b>
8.1	Introducción . . . . .	87
8.2	Recursos utilizados . . . . .	87
8.3	Simulaciones globales . . . . .	88
8.4	Capturas experimentales . . . . .	91
<b>9</b>	<b>Conclusiones y líneas futuras</b>	<b>97</b>
9.1	Conclusiones . . . . .	97
9.2	Líneas futuras . . . . .	98
<b>10</b>	<b>Pliego de condiciones</b>	<b>99</b>
<b>11</b>	<b>Presupuesto</b>	<b>101</b>
	<b>Bibliografía</b>	<b>103</b>



# Índice de figuras

1.1	Modelo Hybrid Power Line/Wireless Channel (HPWC) [2]. . . . .	5
1.2	Modelo Hybrid Power Line/Wireless Single-Relay Channel (HSRC) [2]. . .	5
1.3	Esquema de la solución híbrida utilizando las tecnologías LoRa y PLC [3].	6
1.4	Arquitectura de la tecnología PLC para Smart Grid [5]. . . . .	7
1.5	Sistema MIMO 3x4 de ejemplo [7]. . . . .	9
1.6	Tasa binaria y ganancia de un sistema MIMO en función del país y el número de transmisores y receptores [7]. . . . .	9
1.7	FCS típico con configuración centralizada y los elementos que lo componen [8]. . . . .	10
2.1	Regulaciones para comunicaciones PLC de banda ancha y banda estrecha en varias partes del mundo [11]. . . . .	16
2.2	Desarrollo de los diferentes estándar de comunicaciones PLC de banda estrecha cronológicamente [11]. . . . .	17
2.3	Estándares y entidades reguladoras en función de la banda de frecuencia de banda estrecha [11]. . . . .	17
2.4	Desarrollo de los diferentes estándar de comunicaciones PLC de banda ancha cronológicamente [11]. . . . .	18
2.5	Composición de la tarjeta de evaluación ZC706 como SoC utilizado [12]. . .	19
2.6	Lectura y escritura mediante el bus AXI-Lite [13]. . . . .	20
2.7	Tarjeta de evaluación ZC706 utilizada para el desarrollo del trabajo [14]. .	20
2.8	Diagrama de bloques del transmultiplexor FBMC [15]. . . . .	21
2.9	Diagrama de bloques de la DCT4e [15]. . . . .	22
3.1	Esquema del diseño propuesto implementado en un SoC. . . . .	24
3.2	Interacción del software en el diseño hardware para establecer la configu- ración y funcionalidad del AFE correspondiente. . . . .	28

3.3	Transmisión inicial de las tramas de sincronismo y conmutación de memorias al iniciar la comunicación. . . . .	29
3.4	Arquitectura del driver del AFE. . . . .	30
3.5	Bifurcación de la información recibida del canal PLC para realizar el sincronismo. . . . .	31
3.6	Proceso de sincronismo llevado a cabo en el diseño. . . . .	32
4.1	Diagrama de bloques simplificado del funcionamiento del dispositivo MAX2981 [18]. . . . .	34
4.2	Montaje del AFE en la tarjeta mediante su conector FMC. . . . .	34
4.3	Terminales del MAX2981 utilizado como AFE [18]. . . . .	35
4.4	Pines del MAX2981 utilizado como AFE. . . . .	37
4.5	Cronograma del tiempo de mantenimiento del ADC/DAC que incorpora el AFE [18]. . . . .	38
4.6	Descripción de la escritura de registros internos del AFE dada por el fabricante Maxim Integrated [18]. . . . .	38
5.1	Esquema de las partes que componen el módulo <i>AFE Driver</i> . . . . .	44
5.2	Configuración de los <i>AFE Drivers</i> del diseño. . . . .	45
5.3	Esquema de las partes que componen el módulo <i>AFE Driver</i> transmisor. . . . .	48
5.4	Esquema de las partes que componen el módulo <i>AFE Driver</i> receptor. . . . .	49
5.5	Direcciones de memoria de los módulos <i>AFE Driver</i> . . . . .	50
5.6	Definición de la memoria FIFO que realiza el cambio de dominio de reloj en transmisión. . . . .	51
5.7	Cronograma del funcionamiento de la memoria FIFO en transmisión. . . . .	52
5.8	Definición de la memoria FIFO que realiza el cambio de dominio de reloj en recepción. . . . .	52
5.9	Cronograma del funcionamiento de la memoria FIFO en recepción. . . . .	53
5.10	Arquitectura del módulo SPI integrado en el <i>AFE Driver</i> . . . . .	54
5.11	Configuración del registro 1 del AFE. . . . .	54
5.12	Configuración del registro 2 del AFE. . . . .	55
5.13	Configuración del registro 3 del AFE. . . . .	55
5.14	Configuración del registro 6 del AFE. . . . .	55
5.15	Descripción del método de escritura de los registros del AFE dada por el fabricante. . . . .	56

5.16 Simulación de la escritura de los registros del AFE. . . . .	56
5.17 Simulación de la escritura del registro 1 del AFE en configuración transmisora. . . . .	57
5.18 Simulación de la escritura del registro 1 del AFE en configuración receptora. . . . .	57
5.19 Simulación de la escritura del registro 2 del AFE. . . . .	57
5.20 Simulación de la escritura del registro 3 del AFE. . . . .	57
5.21 Simulación de la escritura del registro 6 del AFE. . . . .	58
5.22 Captura experimental en placa de la escritura de los registros del AFE obtenida mediante el ILA. . . . .	58
5.23 Prueba experimental del SPI en el osciloscopio con la escritura de los tres primeros registros. . . . .	58
5.24 Prueba experimental del SPI en el osciloscopio con la escritura de los dos primeros registros. . . . .	59
5.25 Prueba experimental del SPI en el osciloscopio con la escritura de todos los registros. . . . .	59
6.1 Correlación mediante un producto en el dominio de la frecuencia planteada para el sincronismo del diseño. . . . .	63
6.2 Modelo de la FFT utilizada en el diseño cuyas etapas son todas exactamente iguales [19]. . . . .	64
6.3 Esquema simplificado de la celda DSP48E1 [20]. . . . .	65
6.4 Tramas de sincronismo mediante las dos representaciones. . . . .	66
6.5 Tramas de sincronismo en binario natural a la entrada del AFE empleado. . . . .	67
6.6 Parte real de la FFT de las tramas recibidas por el canal PLC. . . . .	69
6.7 Parte imaginaria de la FFT de las tramas recibidas por el canal PLC. . . . .	69
6.8 Parte real del conjugado de la FFT de las tramas de sincronismo ideales. . . . .	70
6.9 Parte imaginaria del conjugado de la FFT de las tramas de sincronismo ideales. . . . .	70
6.10 Parte real de la correlación cruzada en el dominio de la frecuencia para las distintas representaciones. . . . .	71
6.11 Parte imaginaria de la correlación cruzada en el dominio de la frecuencia para las distintas representaciones. . . . .	71
6.12 Parte real de la correlación cruzada en el dominio de la frecuencia para diferentes cuantificaciones. . . . .	71
6.13 Parte imaginaria de la correlación cruzada en el dominio de la frecuencia para diferentes cuantificaciones. . . . .	71

6.14	Parte real de la correlación en el dominio del tiempo sin retardo en las tramas introducidas. . . . .	72
6.15	Parte real de la correlación en el dominio del tiempo con retardo en las tramas introducidas. . . . .	73
6.16	Determinación de las 1024 muestras entrantes al módulo de sincronismo en función del pico de correlación. . . . .	73
6.17	Arquitectura del módulo de <i>Sincronismo</i> . . . . .	74
6.18	Prueba experimental del módulo de <i>Sincronismo</i> . . . . .	75
7.1	Esquema de la interacción del módulo <i>Interfaz</i> con el resto del diseño. . . .	78
7.2	Esquema del módulo <i>Interfaz</i> en transmisión. . . . .	80
7.3	Cronograma para transmitir las tramas a 50 MHz con un reloj de 100 MHz. . . .	80
7.4	Cronograma del funcionamiento de la conmutación entre memorias. . . . .	81
7.5	Cronograma de la lectura de la FIFO del <i>AFE Driver</i> . . . . .	82
7.6	Cronograma de la escritura de las memorias DUAL PORT. . . . .	82
7.7	Funcionamiento de la conmutación entre memorias de doble puerto. . . . .	83
7.8	Funcionamiento de la conmutación entre memorias de doble puerto. . . . .	83
7.9	Composición en sectores de la memoria FIFO que almacena las muestras recibidas mientras se busca la primera trama de información. . . . .	84
7.10	Envío de tramas al FBMC receptor desde la interfaz. . . . .	85
7.11	Escritura y lectura de la memoria DUAL PORT que proporciona las tramas al FBMC receptor. . . . .	85
8.1	Valores de entrada al sistema simulando el transmisor FBMC. . . . .	89
8.2	Tramas de sincronismo y de datos a la salida del módulo <i>Interfaz</i> transmisor. . . .	89
8.3	Salida del driver transmisor del AFE. . . . .	89
8.4	Salida de las tramas del driver receptor con un canal PLC ideal. . . . .	90
8.5	Bloques de 1024 muestras introducidos al módulo de <i>Sincronismo</i> . . . . .	90
8.6	Pico de correlación obtenido del módulo de <i>Sincronismo</i> . . . . .	90
8.7	Salida del módulo <i>Interfaz</i> de recepción. . . . .	91
8.8	Captura experimental de la salida del transmisor FBMC. . . . .	92
8.9	Captura experimental de las tramas de sincronismo y de datos. . . . .	92
8.10	Captura experimental de las tramas de sincronismo transmitidas al canal mediante el AFE transmisor. . . . .	92

---

8.11	Captura experimental del espectro de la señal transmitida por el AFE al canal PLC. . . . .	93
8.12	Captura experimental de la entrada del driver transmisor y de la salida del driver receptor, tras pasar por el canal PLC. . . . .	93
8.13	Captura experimental de la entrada al módulo de <i>Sincronismo</i> . . . . .	93
8.14	Captura experimental de la salida del módulo de <i>Sincronismo</i> con el pico de correlación correspondiente. . . . .	94
8.15	Captura experimental de la comparativa entre la señal recibida y la transmitida. . . . .	94
8.16	Captura experimental de la salida del receptor FBMC. . . . .	95
8.17	Montaje realizado mediante un único SoC. . . . .	95
8.18	Montaje realizado mediante dos SoC. . . . .	96





# Índice de tablas

4.1	Descripción de los pines del AFE. . . . .	36
4.2	Terminales de los conectores FMC para conectar a los puertos del diseño de la arquitectura. . . . .	37
4.3	Direcciones de los registros del AFE. . . . .	39
4.4	Configuración del registro 1 del AFE. . . . .	39
4.5	Configuración del registro 2 del AFE. . . . .	40
4.6	Configuración del registro 3 del AFE. . . . .	40
4.7	Configuración del registro 6 del AFE. . . . .	41
6.1	Ejemplo de ordenación de las muestras en el modelo de FFT escogido para 8 puntos. . . . .	64
6.2	Conversión de la información a binario natural para enviarla al conversor DA. . . . .	67
6.3	Características de la cuantificación de las tramas de sincronismo en la entrada/salida del AFE empleado. . . . .	67
6.4	Determinación de los bits de parte entera a la salida de la FFT de las tramas recibidas. . . . .	68
6.5	Características de la cuantificación a la salida de la FFT de las tramas de sincronismo recibidas. . . . .	68
6.6	Determinación de los bits de parte entera del conjugado de la FFT de las tramas ideales. . . . .	69
6.7	Características de la cuantificación del conjugado de la FFT de las tramas de sincronismo ideales . . . . .	70
6.8	Determinación de los bits de parte entera de la correlación. . . . .	70
6.9	Características de la cuantificación a la salida de la correlación en el dominio de la frecuencia. . . . .	71
6.10	Determinación de los bits de parte entera de la correlación en el dominio del tiempo. . . . .	72

6.11	Características de la cuantificación de la correlación en el dominio temporal.	72
8.1	Recursos consumidos por la arquitectura propuesta. . . . .	88
11.1	Coste material utilizado en el trabajo (CD). . . . .	101
11.2	Coste de mano de obra en el trabajo (CD). . . . .	102
11.3	Otros conceptos incluidos en el coste. . . . .	102
11.4	Coste total del trabajo incluyendo el IVA. . . . .	102

# Lista de acrónimos

AD	Analog-Digital.
ADC	Analog-Digital Converter.
AFE	Analog Front-End.
ARIB	Association of Radio Industries and Businesses.
AXI	Advanded eXtensible Interface.
BPSK	Binary Phase Shift Keying.
BRAM	Block Random Access Memory.
BUFG	Global Clock Buffer.
CD	Coste Directo.
CI	Coste Indirecto.
CSS	Complementary Set of Sequences.
DAC	Digital-Analog Converter.
DCT	Discrete Fourier Transform.
DSP	Digital Signal Processor.
EN	European Norm.
FBMC	Filter Bank Multicarrier.
FCC	Federal Communications Commision.
FCS	Fligth Control System.
FF	Flip-Flop.
FFT	Fast Fourier Transform.
FIFO	First-Input First-Output.
FMC	FPGA Mezzanine Card.
FPGA	Field-Programmable Gate Array.
FSK	Frequency Shift Keying.

HDL	Hardware Description Language.
HLS	High Level Synthesis.
HPC	High Pin Count.
HPWC	Hybrid Power Line/Wireless Channel.
HSRC	Hybrid Power Line/Wireless Single-Relay Channel.
HW	Hardware.
ICI	Inter-Carrier Interference.
IFFT	Inverse Fast Fourier Transform.
ILA	Integrated Logic Analyzer.
IO	Inputs and Outputs.
IoT	Internet-of-Things.
IPIF	IP Interface.
LD	Line Driver.
LPC	Low Pin Count.
LUT	Look-up Tables.
LUTRAM	RAM built for Look-Up Tables.
MCM	Multi-Carrier Modulation.
MIMO	Multi-Input Multi-Output.
MMCM	Mixed Mode Clock Manager.
OFDM	Orthogonal Frequency Division Multiplexing.
PL	Programmable Logic.
PLC	Power Line Communications.
PRIME	Powerline Related Intelligent Metering Evolution.
PS	Processing System.
S-FSK	Spread Frequency Shift Keying.
SDK	Software Development Kit.
SoC	System-on-Chip.
SPI	Serial Peripheral Interface.
SS	Spread Spectrum.

---

SW Software.



# Memoria





# Capítulo 1

## Introducción

### 1.1 Finalidad del proyecto

Antes de comenzar la descripción detallada de todo el trabajo realizado, es importante establecer la propuesta dentro de un marco de implantación y de desarrollo. Por ello, se comienza en este capítulo inicial realizando una visión general dentro del contexto que se procede a desarrollar. La finalidad de este trabajo radica en realizar comunicaciones de banda ancha y alta velocidad, por canales PLC, dentro de entornos aeroespaciales o vehiculares. Se trata de una importante línea de trabajo e investigación, debido a la presencia de dicho canal en una gran cantidad de lugares donde, tradicionalmente, sirve exclusivamente para lo que se desarrolló en un principio, y, para lo que se utiliza mayoritariamente en la actualidad: proveer de alimentación para el funcionamiento de las diversas aplicaciones y dispositivos. También existen muchos elementos que, sin estar conectados a la red eléctrica, disponen de esta red de manera interna para alimentar o conectar los diferentes dispositivos dentro de un mismo elemento, como resulta en este caso, dentro de los diferentes vehículos/elementos aeroespaciales. Esto conlleva poder realizar comunicaciones por un enlace existente, con el consiguiente ahorro en infraestructura. Para llevar a cabo esta propuesta, es necesario un elemento que realice las etapas de conversión DA/AD, aislamiento y de amplificación. Para todo ello, se utilizan los dispositivos Analog Front-End, que suponen una integración de todos los elementos necesarios en uno único, tanto para la transmisión como para la recepción. Esto provoca una reducción de esfuerzo en el diseño de toda la etapa intermedia entre el diseño y el canal PLC.

Debido a las dificultades que presenta el canal, y que serán recogidas en el siguiente capítulo donde se abordará el canal PLC en detalle, es necesario utilizar modulaciones MCM. Como técnica de acceso al medio, se utilizará por tanto una de las recomendadas por el estándar IEEE para comunicaciones PLC de banda ancha, como es FBMC. La transición entre las tramas de datos enviadas/recibidas por el transmultiplexor FBMC y los AFE será lo que se acometa en este trabajo: dos drivers para configurar los AFE en

el modo que sea requerido, una interfaz para ajustar la técnica de acceso al medio a las características del AFE y un módulo de *Sincronismo* con el cual determinar el inicio de la transmisión de datos en el bloque receptor. En este trabajo no se abordará la estimación e igualación de canal.

## 1.2 Contexto

El planteamiento de este trabajo se engloba en la nueva generación de comunicaciones de banda ancha, al utilizar una técnica de acceso al medio planteada inicialmente para las comunicaciones 5G, ya que FBMC es una evolución de OFDM, utilizada en la generación de comunicaciones de 4G. El canal establecido para estas comunicaciones es la red eléctrica, por lo que se evita desplegar una nueva infraestructura que proporcione la comunicación. Esto provoca una gran ventaja en un nuevos campos de desarrollo, como es el Internet-of-Things (IoT) o las Smart Grids. El IoT se puede definir como una red de dispositivos que se comunican entre sí, para obtener información propia y del resto de elementos conectados, y en consecuencia actuar de manera autónoma. Esto provoca la dotación de inteligencia a los dispositivos, capaces de actuar en función de la información del entorno y del resto de elementos, y de esta manera tener una mayor independencia. La interacción entre estos dispositivos supone un problema en la actualidad, siendo la vía inalámbrica un canal demasiado inseguro para transmitir información, y dar pie a la ciberdelincuencia. El uso del canal PLC puede implicar un aumento de la seguridad y una reducción de costes. El IoT es un término muy amplio que engloba toda esta tecnología, pero es escalable desde una casa a una ciudad. Por otro lado, las Smart Grids se pueden definir como las redes eléctricas dotadas de inteligencia, y de esta manera mejorar la eficiencia energética, y en principio, reducir los costes en el consumidor, además de reducir el consumo y la contaminación en las fuentes que no son limpias.

En [1] se presenta un trabajo, en el cual se pretende dar cabida a la conexión de todos los dispositivos que compondrían el IoT utilizando el canal PLC, y además en este ámbito de manera destacada debido a que muchos dispositivos requieren la conexión a la red eléctrica para el suministro de energía. El trabajo menciona que debido a la aparición de dispositivos más avanzados, como es el SoC utilizado, se pueden llevar a cabo técnicas de modulación más sofisticadas, como es la FBMC establecida en este trabajo. No solo habla del IoT, también hace referencia a las smart homes, smart citys, con electrodomésticos capaces de regular su consumo de manera autónoma en función de parámetros externos obtenidos de la red.

En [2] se presenta un trabajo en donde busca una solución híbrida entre comunicaciones inalámbricas y comunicaciones PLC para IoT y Smart Grids. Hacen hincapié en la ne-

cesidad de buscar la red comunitaria entre todos y una tecnología capaz de aportar la velocidad suficiente para las necesidades de estas aplicaciones. También se aborda la temática de la seguridad, muy importante a la hora de desplegar cualquier tecnología o red. Debido en un gran porcentaje al tamaño de los dispositivos conectados dentro de la red del IoT, la seguridad que se puede aplicar a las comunicaciones no puede ser muy avanzada o compleja, por lo que supone una vulnerabilidad grave dentro de esta tecnología. La intromisión en dispositivos que pueden estar dentro de una red domestica normal puede suponer perjuicios pero de manera moderada; no sería así en dispositivos de un hospital, donde elementos clínicos suministran medicamentos en función de otros parámetros obtenidos, ahí la seguridad es vital. Es, por tanto, un parámetro primordial en el desarrollo de estas tecnologías. En la figura 1.1 se representa el primer modelo que se plantea, donde el transmisor emite la señal hacia el destino por ambos canales, el inalámbrico y el PLC. En la figura 1.2 se muestra el segundo modelo planteado, que utiliza el mismo modelo anterior pero además en paralelo se manda a un nodo intermedio.

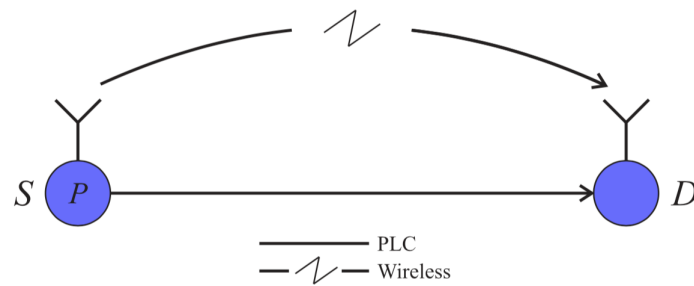


Figura 1.1: Modelo HPWC [2].

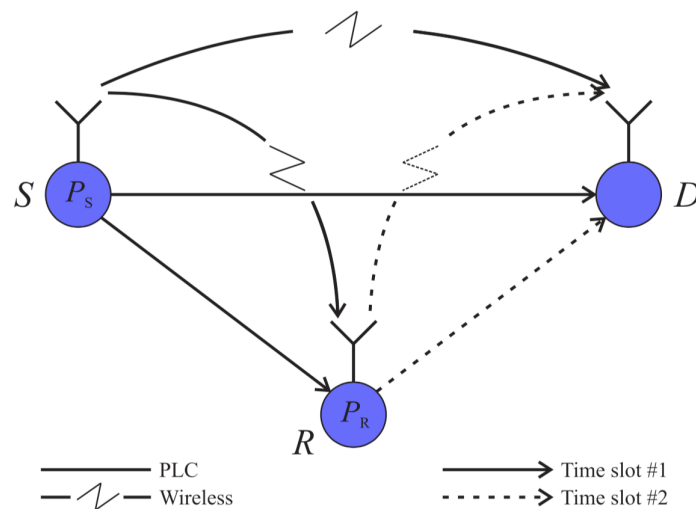


Figura 1.2: Modelo HSRC [2].

En [3] se presenta una solución híbrida entre LoRa y PLC para una red de sensores con

distancias más elevadas de lo que es habitual encontrar en trabajos de IoT. LoRa es un tipo de tecnología para comunicaciones inalámbricas cuyo radio de acción es bastante superior al resto de competidores. Vuelve a ser igual que en [2], una solución híbrida entre comunicaciones inalámbricas y PLC. En la figura 1.3 se muestra cómo se combinan ambas tecnologías, aparece representado con una antena la parte de la comunicación realizada de manera inalámbrica con LoRa y mediante un enchufe la parte de comunicación PLC. Entre medias, un modem de cada tecnología y los ajustes necesarios para pasar de una tecnología a la otra.

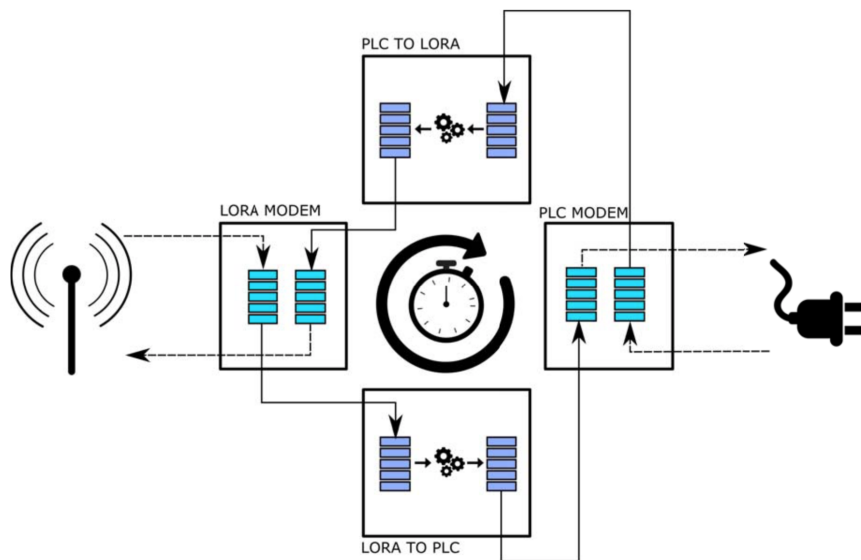


Figura 1.3: Esquema de la solución híbrida utilizando las tecnologías LoRa y PLC [3].

En [4] se presenta un trabajo sobre Smart Grids mediante comunicaciones PLC en la última etapa de la red. Se basa en la realización de modelos y topologías óptimas para favorecer esta tecnología en el futuro.

En [5] se introduce un trabajo, en el cual se presenta la implementación y una arquitectura para Smart Grids mediante canales PLC en una empresa de energía de gestión privada. Presentan modelos para poder utilizarse en banda ancha y banda estrecha, centrándose en la banda estrecha debido a que se adapta mejor a la lectura de contadores que pretende incorporar la compañía en todas las viviendas o lugares que presentan su tendido eléctrico. La idea de este trabajo es que en un futuro se pueda extrapolar para empresas de servicio público sirviendo como base la red de una gran empresa. En la figura 1.4 se representa una hipotética ciudad, en la que, hasta llegar al hogar con una estandarización PRIME, pasa por subestaciones primarias y secundarias dividiendo las casas en sectores de las que se ocupa cada subestación secundaria. Un conjunto de subestaciones forman una red local PLC.

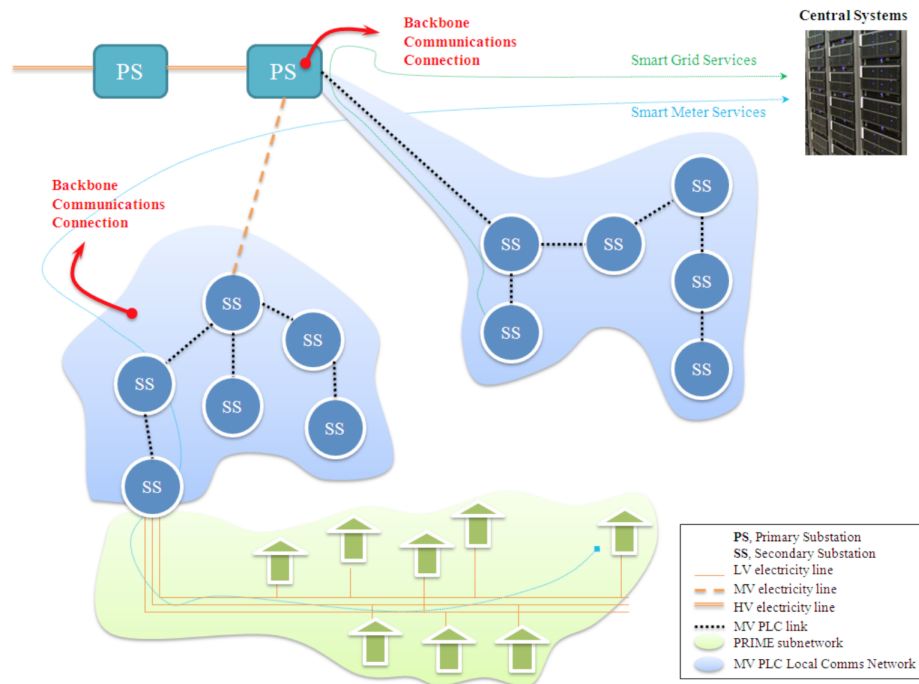


Figura 1.4: Arquitectura de la tecnología PLC para Smart Grid [5].

### 1.3 Entornos de desarrollo

La tarjeta, en donde se implementa todo el diseño abordado en este trabajo y a la que se conectan los AFE para interactuar con el canal PLC, pertenece al fabricante Xilinx Inc., por lo que se utilizarán la gama de productos y herramientas que ofrece para su diseño. El diseño de gran parte de la arquitectura se acomete en el programa Vivado (evolución del ISE para los últimos dispositivos sacados por el fabricante) con el lenguaje de descripción hardware VHDL. En esta misma herramienta, se puede realizar tanto la síntesis e implementación, como las simulaciones funcionales y temporales del diseño. La parte de la arquitectura que no se acomete con lenguaje de bajo nivel, como es el sincronismo, con el cual se determina el inicio de la transmisión de datos por parte del transmisor FBMC, se realiza mediante la herramienta Vivado High Level Synthesis (HLS), con la que mediante un lenguaje de alto nivel como es C, se pueden realizar módulos hardware descritos en lenguaje de bajo nivel. Esta herramienta puede suponer una disminución en el tiempo de desarrollo principalmente en diseños basados en iteraciones, como es el caso del sincronismo planteado y que se desarrolla en detalle en capítulos posteriores. Para hacer el diseño flexible y reconfigurable, se utilizará uno de los procesadores de ARM que incorpora el SoC utilizado. Para ello, se realiza una aplicación programada en C mediante la herramienta Xilinx SDK. El procesador y la arquitectura interactúan a través de los buses específicos de ARM como son los Advanced eXtensible Interface (AXI). Existen varios tipos diferenciados, en uno la información se muestra como un flujo continuo de datos, y el otro mediante el mapeado de memoria; el elegido para esta aplicación, en

su especificación AXI-Lite. En el siguiente capítulo se profundiza en estos sistemas que disponen de parte hardware y software comunicados por dichos buses.

Antes de acometer el diseño hardware, es necesario realizar un modelo en coma fija para probar que la arquitectura se ajusta a los resultados buscados. Para ello, se hará uso de la herramienta Matlab, que mediante el uso de funciones específicas, permiten fijar el tamaño de palabra, repartiendo los bits de la misma entre la parte entera y la parte decimal. Debido a que se producen operaciones en coma fija dentro del diseño, será necesario hacer un estudio de la operación en cuestión y solventar los posibles desbordamientos si se pretende tener un modelo lo más fiel posible y con menor error con respecto al que sería en coma flotante.

## 1.4 Aplicaciones aeroespaciales o vehiculares

Las comunicaciones PLC se utilizan en una gran cantidad de aplicaciones dentro de entornos aeroespaciales. En este apartado se presentan varios artículos que utilizan dichas comunicaciones para distintas aplicaciones y existen muchas líneas de investigación abiertas.

En [6] se presenta un trabajo, en el cual se realiza una caracterización y modelado del canal PLC para utilizar las técnicas Multi-Input Multi-Output (MIMO) en las comunicaciones PLC internas de las aeronaves. Esta caracterización supone siempre un problema importante en este tipo de comunicaciones y son necesarios trabajos de este tipo. Estas técnicas siempre se han asociado a múltiples antenas transmitiendo y recibiendo, como mejora u optimización de las comunicaciones inalámbricas; en este caso son varios enlaces por los que se transmite. La idea principal se basa en establecer un determinado número de transmisores y de receptores, y se suelen establecer dos estrategias típicamente: todas las antenas transmiten la misma información, por lo que si en recepción se pierde algo de una, la obtienen por otro lado. Esto reduce la probabilidad de error de manera considerable ya que se manda la información de manera redundante; la otra técnica consiste en dividir la información en partes y transmitir cada una de ellas por los diferentes transmisores, esto provoca un aumento de la tasa binaria. MIMO está presente en dos estándares de comunicaciones PLC. En [7] se presenta también un trabajo que utiliza la técnica MIMO en comunicaciones PLC y aporta una gran cantidad de pruebas que validan la mejora que implica en las comunicaciones en todo lo referente a lo comentado. En la figura 1.5 se puede ver cómo funciona esta técnica con varios transmisores y receptores, y en la figura 1.6 se presenta, en función de los diferentes estándares de los países, la mejora correspondiente en la tasa binaria en función del números de transmisores/receptores.

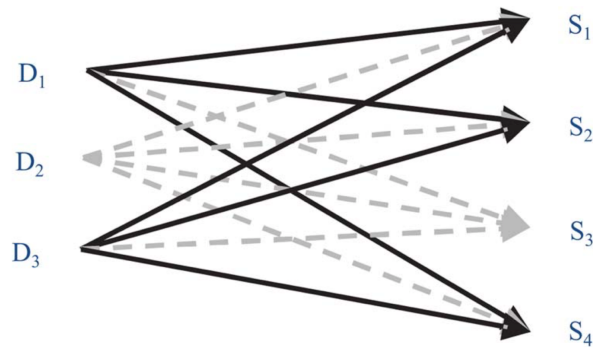


Figura 1.5: Sistema MIMO 3x4 de ejemplo [7].

MIMO configuration	EU mask		US mask		JP mask	
	Mbit/s	gain	Mbit/s	gain	Mbit/s	gain
1 × 1	82		62		6	
1 × 2	126	1.55	103	1.65	23	3.63
1 × 4	173	2.12	143	2.30	34	5.37
2 × 2	153	1.87	121	1.94	23	3.52
2 × 4	235	2.88	190	3.05	41	6.35

Figura 1.6: Tasa binaria y ganancia de un sistema MIMO en función del país y el número de transmisores y receptores [7].

En [8], se utilizan las comunicaciones PLC para el Fligh Control System (FCS). Los FCS que se utilizan en las aeronaves modernas requieren con el paso del tiempo un mayor ancho de banda en las comunicaciones con los subsistemas que proporcionan información de vuelo debido a que cada vez aumenta la complejidad de dichos subsistemas, por lo que suele ser necesario hasta una tasa de 20 Mbps. En la figura 1.7 se presenta un FCS centralizado con todos los elementos que lo componen.

En [9] se presenta un trabajo, en el cual se establecen varios escenarios, como son: un coche y un barco; con el fin de implementar comunicaciones PLC de banda ancha disminuyendo el peso y el coste de la infraestructura. Aunque no sea un dato que sea muy conocido por la mayoría de personas, el peso del cableado de cualquier vehículo supone el tercer elemento de mayor peso, después del motor y la caja de cambios. Con la evolución del sector hacia las energías renovables y funcionamiento mediante baterías, es de vital importancia reducir este peso para poder ser así más eficiente, y por tanto obtener una mayor autonomía sin volver a recargar las baterías.

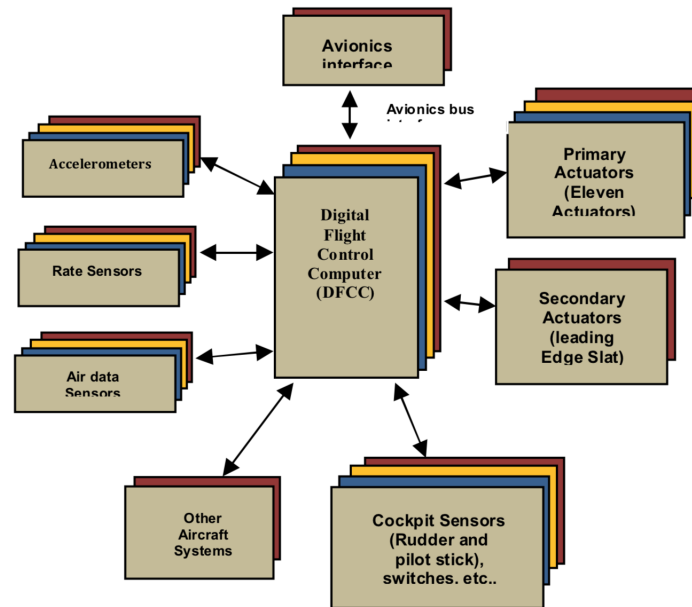


Figura 1.7: FCS típico con configuración centralizada y los elementos que lo componen [8].

## 1.5 Estructura del documento

En este apartado se realiza una descripción de los próximos capítulos que presenta el documento. La línea seguida para la descripción de este trabajo busca inicialmente ubicar el proyecto dentro de una posible alternativa de canal de comunicación, que está presente en muchos elementos (como pueden ser aeronaves o vehículos), y todos los elementos necesarios para que se produzcan. Después pasar a describir la arquitectura global propuesta para realizar estas comunicaciones, y de manera individual de todo lo que la compone, quedando organizado de la siguiente manera.

En el capítulo 2 se describen las comunicaciones PLC, junto con los dos tipos que se establecen en función del ancho de banda de la información transmitida: banda ancha y banda estrecha. Se describen además los estándares que existen en la actualidad para cada una de ellas, y si se ajustan a lo que se plantea y a las limitaciones que presentan los conversores. La tecnología utilizada para llevar a cabo el trabajo son los SoC, por ello, se describen en detalle y se explica el porqué se están convirtiendo en una alternativa a la hora de acometer diseños en tiempo real y las diferencias con las FPGAs tradicionales. Por último, se describen los trabajos previos realizados en la temática PLC utilizando SoCs y que sirven de base a este trabajo para poder realizar su finalidad.

En el capítulo 3 se realiza una descripción de todo el diseño propuesto, en donde se puede diferenciar una parte software y una parte hardware. Se establece qué parte del diseño se distribuye en cada lugar para después explicar en detalle en el resto de capítulos los



diferentes módulos. Es muy importante dar este enfoque global antes de la descripción para poder ubicar a la perfección cada bloque dentro del diseño y no perderse en la definición. La idea es dentro de cada módulo hacer una descripción cronológica, por lo que si los módulos aparecen tanto en el bloque receptor como transmisor, comenzar siempre por la parte transmisora para terminar en la receptora.

En el capítulo 4 se describen detalladamente las características del AFE utilizado, y que resultan más relevantes y de consideración a la hora de realizar el diseño de la arquitectura de su driver, explicado en el capítulo 5. Para ello, se describen las etapas que componen el AFE, para entender mejor porque el driver necesita configurar ciertas señales para darle una determinada funcionalidad. Se describirán todas esas señales y la función de configuración que representan. Por último, se describirán los registros internos que disponen estos dispositivos y que, según el valor proporcionado mediante una interfaz serie, se pueden modificar para conseguir características distintas del AFE.

En el capítulo 5 se describe la arquitectura del driver encargado de controlar y gestionar el AFE. Este dispositivo cuenta con las dos funcionalidades y los dos conversores AD/DA, por lo que será necesario establecer dos drivers, uno que configure uno como transmisor y otro como receptor. Será por tanto una de las funciones clave de este módulo, la configuración de cada dispositivo. Para hacer genérico y flexible el diseño, el driver de transmisión y de recepción presentan la misma arquitectura, pero se configuran de manera diferente vía software para cada uno de los AFE.

En el capítulo 6 se describe la arquitectura de sincronismo empleada en la propuesta del diseño. En el capítulo 2, se describirán los trabajos anteriormente realizados en relación al sincronismo dentro de las comunicaciones PLC, de donde surge la base teórica de este desarrollo. En el capítulo 6, se describe inicialmente la base teórica del método, que es la realización de la correlación cruzada en el dominio de la frecuencia. Se estudiarán las operaciones realizadas y se buscará la manera más óptima de realizar las transformaciones. Después se expone un modelo en coma fija emulando todas las operaciones que conlleva y se comprueban los errores y desbordamientos en función de las operaciones realizadas. Obtenido el modelo en coma fija, se expone la arquitectura con las variantes introducidas para su mejor diseño e implementación.

En el capítulo 7 se realiza la descripción de la arquitectura de la interfaz, encargada de interactuar con todos los módulos del diseño. Se puede desglosar en una parte dedicada al bloque transmisor y otra al receptor. En la parte transmisora, se encarga de realizar un acondicionamiento de las tramas de datos que proporciona el transmisor FBMC. Este

ajuste se basa en adaptarlas a un formato correcto para la justa interpretación de los conversores o la cuantificación del tamaño de palabra para ajustarse al bus de datos del AFE, entre otras. También se encarga de albergar las tramas de sincronismo que se enviarán en el comienzo de la comunicación antes de las tramas de datos para determinar el inicio. Implementará también una lógica de control que permita gestionar anchos de banda superiores a los limitados por el AFE. En la parte receptora, se encargará de la lógica de control para un consumo de datos superior al producido, interactuará con el sincronismo para determinar cuál es la primera subportadora de la primera trama y se ajustará a las exigencias del receptor FBMC.

En el capítulo 8 se presentan los resultados obtenidos que validan la propuesta de diseño realizada. Inicialmente, se presentan los recursos consumidos por la arquitectura frente a los disponibles en la tarjeta utilizada. Después, se expondrán simulaciones de la arquitectura intermedia encargada de la gestión de todos los módulos para después, exponer simulaciones incluyendo el canal PLC y el transmultiplexor FBMC.

# Capítulo 2

## Antecedentes

### 2.1 Comunicaciones PLC (Power Line Communication)

Las comunicaciones PLC son aquellas que se realizan a través de la red eléctrica; esto permite realizar una reducción de las redes de comunicación presentes y futuras debido a que se utiliza la infraestructura ya existente, pudiendo alcanzar lugares en los que las nuevas generaciones de redes aún no se han implementado y sin la suficiente demanda tampoco lo harán en el futuro. Sin embargo, gran parte de las dificultades que presenta PLC se deben a la complejidad del canal, no creado inicialmente para la transmisión de información. Otro de los problemas que presenta es el ruido, que es una combinación de varios tipos. Entre ellos, se pueden distinguir los siguientes, descritos en [10]:

- Ruido coloreado de fondo: se produce por las contribuciones de varias fuentes de ruido de baja potencia.
- Ruido de banda estrecha o interferencia de banda estrecha: se produce por señales sinusoidales con modulación en amplitud.
- Ruido impulsivo periódico asíncrono: se produce por las fuentes de alimentación conectadas a la red.
- Ruido impulsivo periódico síncrono: se produce por las fuentes de alimentación conectadas a la red.
- Ruido impulsivo asíncrono: es producido por transitorios rápidos de la red.

Para transmisiones de banda ancha, y teniendo en cuenta las dificultades del canal, habitualmente se emplean modulaciones multi-portadora, tales como OFDM o FBMC, que minimizan el desvanecimiento selectivo en frecuencia; así como diversas técnicas de sincronismo, y estimación e igualación de canal. En este trabajo, se acomete el sincronismo pero no la estimación e igualación de canal.

Se puede distinguir entre dos tipos de comunicaciones PLC en función del ancho de banda que requiera:

- Comunicaciones PLC de banda estrecha.
- Comunicaciones PLC de banda ancha.

### 2.1.1 Regulación de las comunicaciones PLC

La gran cantidad de elementos que componen e integran la red eléctrica en sus diferentes aplicaciones ha provocado la necesidad de realizar las regulaciones pertinentes con los consiguientes estándares desarrollados por las diferentes entidades reguladoras. Estas regulaciones facilitan la convivencia dentro de la red de los elementos de comunicación y los que únicamente utilizan la red como fuente de alimentación. Por tanto, los sistemas PLC deben de seguir una serie de normas al estar conectadas a la red.

Como ya se ha comentado, se puede hacer una clasificación de los sistemas PLC en función de la banda de frecuencia a la que trabajen. Por tanto, los sistemas en la banda 3-500 kHz serán sistemas de banda estrecha y de 1.8-100 MHz serán sistemas de banda ancha. Las comunicaciones PLC de banda ancha, al realizarse a elevadas frecuencias, provocan interferencia electromagnética perjudicial para el resto de elementos, por lo que la regulación entre banda ancha y estrecha será diferente. Todo lo referente a regulación y estandarización explicado en este capítulo se obtiene en [11], en donde aparece una explicación más detallada.

#### 2.1.1.1 Regulación de las comunicaciones PLC de banda estrecha

Un estándar regulatorio a nivel europeo de las comunicaciones PLC de banda estrecha es el European Norm (EN) 50065. La EN distingue cuatro bandas de frecuencia:

- CELENEC-A: 3-95 kHz
- CELENEC-B: 95-125 kHz
- CELENEC-C: 125-140 kHz
- CELENEC-D: 140-148.5 kHz

En esta norma europea, se especifican los límites de emisión en términos de: nivel de máximo voltaje y procedimientos de medida. La banda CENELEC-A la utilizan las compañías eléctricas, la banda CENELEC-B-D se puede utilizar solo en instalaciones comerciales y en CENELEC-C hay que utilizar obligatoriamente CSMA/CA.

No existe un estándar armonizado para las frecuencias superiores de banda estrecha en el rango de 150-500 kHz, por tanto se fija el estándar IEEE 1901.2, estableciendo los límites de distorsión máxima en función de la densidad espectral de frecuencia y los métodos de medida.

En EEUU, las comunicaciones PLC son reguladas por el Federal Communications Commission (FCC). Esta entidad reguladora permite utilizar la tecnología PLC en la banda de frecuencias 9-490 kHz sin protección e interferencia. En Japón, la entidad reguladora Association of Radio Industries and Businesses (ARIB) permite comunicaciones PLC en la banda 10-450 kHz. Como se puede apreciar, en todas las regiones comentadas regulan las comunicaciones PLC de banda estrecha por debajo de los 500 kHz.

#### 2.1.1.2 Regulación de las comunicaciones PLC de banda ancha

Como ya se ha comentado, las comunicaciones PLC de banda ancha suponen un problema mayor que las de banda estrecha debido a la alta frecuencia de funcionamiento, provocando interferencia a la línea eléctrica. La fijación de los límites de emisión se ve complicado por el hecho de obtener esta nueva funcionalidad de la red eléctrica: las comunicaciones de alta velocidad. Es por tanto necesario realizar una distinción entre los dos tipos de utilidades en función del tipo de dispositivo conectado. Esta problemática aparece tratada en la EN 50561-1, aplicándose a la banda frecuencial en el rango 1.6-30 MHz y distinguiendo fuente de alimentación y comunicaciones. Además, el estándar exige un control dinámico de potencia.

En EEUU, el 47 CFR 15 define las comunicaciones PLC de banda ancha para el hogar. Establece una banda de frecuencias de funcionamiento entre 1.705-80 MHz. En la figura 2.1 se puede ver una tabla con un resumen de todo lo comentado sobre regulación de comunicaciones PLC en varios países, tanto para banda estrecha como para banda ancha.

#### 2.1.2 Comunicaciones PLC de banda estrecha

Las comunicaciones PLC de banda estrecha son aquellas que se encuentran en frecuencias inferiores a los 500 kHz con regímenes binarios no superiores a los 200 kbps. Con este ancho de banda, dichas comunicaciones se utilizan para aplicaciones que no requieran transmisiones de gran volumen de datos, como puede ser la lectura de contadores, domótica, monitorización de la red, y demás aplicaciones que no exijan altos requerimientos de transmisión. Las modulaciones que se utilizan para estas comunicaciones son las siguientes: Frequency Shift Keying (FSK), Spread Frequency Shift Keying (S-FSK), Binary Phase Shift Keying (BPSK), Spread Spectrum (SS), OFDM, FBMC.

Region	Standard/ Regulation	Remarks
Narrowband PLC		
Europe	EN 50065 3–148.5 kHz	CENELEC A band for utility use CENELEC B-D bands for consumer use CSMA/CA in CENELEC C band
	IEEE 1901.2 148.5–500 kHz	Not a European Harmonized Standard
USA	47 CFR §15 9–490/500 kHz	Rules for power line carrier or carrier current systems apply
Japan	ARIB STD T-84 10–450 kHz	CSMA/CA required
Broadband PLC		
Europe	EN 50561-1 1.6065–30 MHz	Dynamic power control Static and dynamic notching
USA	47 CFR §15 1.705–80 MHz	Subpart G for access BB PLC Interference mitigation and avoidance Excluded bands and zones

Figura 2.1: Regulaciones para comunicaciones PLC de banda ancha y banda estrecha en varias partes del mundo [11].

#### 2.1.2.1 Estándares para comunicaciones PLC de banda estrecha

Dentro de las comunicaciones de banda estrecha, el término HDR significa tener tasas de cientos de kbps en el rango frecuencial 3-500 kHz. Para la industria, existen tres especificaciones: Powerline Related Intelligent Metering Evolution (PRIME) y G3-PLC, el ITU-T que establece las recomendaciones ITU-T G.9955 y G.9956 para establecer el estándar de capa física y de enlace, y el IEEE que establece su estándar IEEE 1901.2 ya mencionado. En la figura 2.2 se puede ver cómo las tres entidades comentadas han establecido los diferentes estándares a lo largo del tiempo.

En la figura 2.3 se representa el rango de frecuencias disponible para comunicaciones PLC de banda estrecha, representando los diferentes estándares junto con las entidades reguladoras para la zona geográfica correspondiente, y los tramos frecuenciales dentro de las mismas.

#### 2.1.3 Comunicaciones PLC de banda ancha

Las comunicaciones PLC de banda ancha son aquellas que se encuentran en el rango de frecuencia por encima de los 2 MHz y que se utilizan para transmisiones de alta velocidad, llegando a obtener transmisiones con las nuevas modulaciones multi-portadora de cientos de Mbps. Estas modulaciones multi-portadora utilizadas son OFDM y FBMC.

En el capítulo 1 ya se han comentado diversas aplicaciones que requieren banda ancha para sus comunicaciones como puede ser el IoT, para la interacción entre dispositivos, las

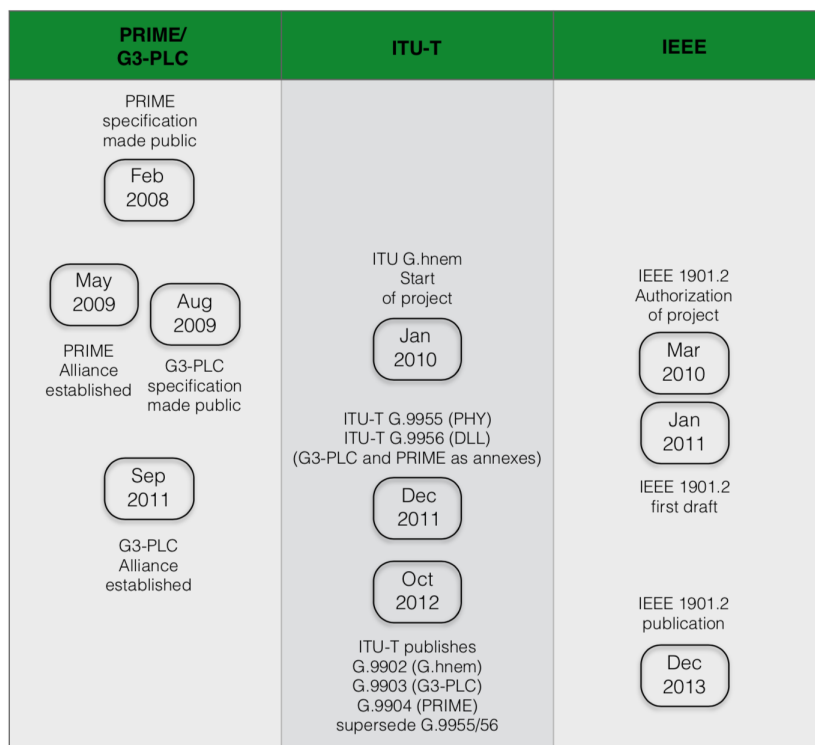


Figura 2.2: Desarrollo de los diferentes estándar de comunicaciones PLC de banda estrecha cronológicamente [11].

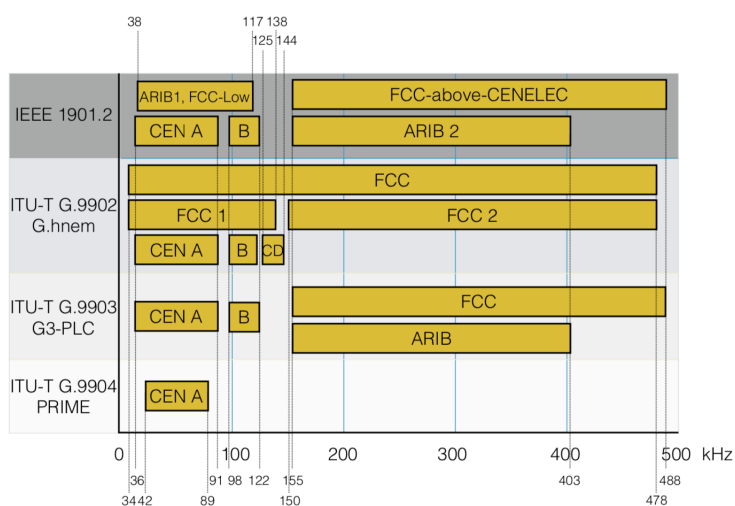


Figura 2.3: Estándares y entidades reguladoras en función de la banda de frecuencia de banda estrecha [11].

Smart Grids o las comunicaciones internas de aeronaves o vehículos.

### 2.1.3.1 Estándares para comunicaciones PLC de banda ancha

Existen varios estándares para comunicaciones PLC de banda ancha, esto provoca la no interoperabilidad entre sistemas PLC y limita la consolidación de la tecnología. El comienzo de este asentamiento comenzó con las entidades IEEE y ITU-T. Las diferencias entre ambos radican en la particularización de ITU para redes domesticas frente a IEEE que está pensado también para el acceso, donde ambos estándares son interoperables. El estándar IEEE 1901, que presenta un mayor interés para este trabajo, tiene una banda de frecuencia entre 2-31.25 MHz y como modulación multi-portadora recomienda OFDM o FBMC. En la figura 2.4 se presenta una línea temporal del desarrollo de los diferentes estándares de comunicaciones PLC de banda ancha.

- IEEE 1901
  - 1901.1-2018: hasta los 12 MHz
  - 1901-2010: hasta los 31.25 MHz

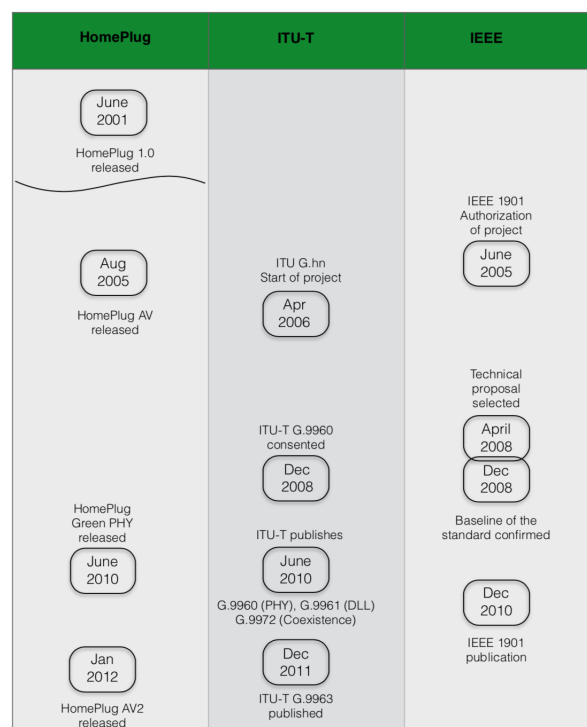


Figura 2.4: Desarrollo de los diferentes estándar de comunicaciones PLC de banda ancha cronológicamente [11].



## 2.2 Sistemas SoC (System-On-Chip)

Las técnicas de acceso al medio, aún mejorando las prestaciones de las comunicaciones PLC, suelen conllevar un aumento considerable de la complejidad computacional. Esto provoca la aparición de una nueva cuestión a analizar, centrada en cómo acometer la implementación en tiempo real de las posibles propuestas. Una posible reciente alternativa es el empleo de SoC para el diseño de estas arquitecturas. Éstos cuentan con una parte lógica programable Programmable Logic (PL), en la que se acomete el diseño de elementos hardware en cualquier HDL Hardware Description Language (HDL), junto con uno o varios procesadores Processing System (PS), donde se puede abordar parte de las tareas mediante un lenguaje de descripción clásico, como C. En la figura 2.5 se puede ver las dos partes junto con los elementos que componen cada una de ellas.

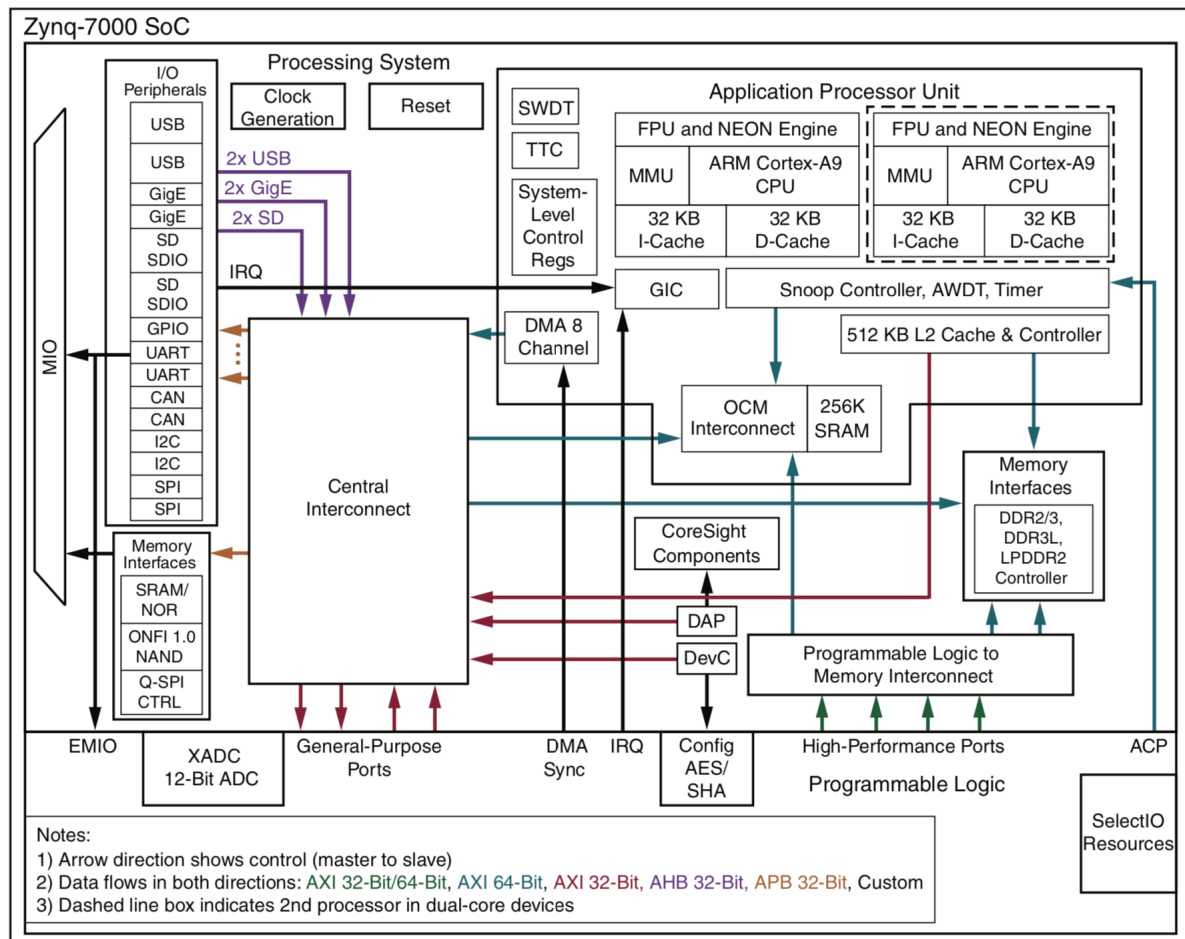


Figura 2.5: Composición de la tarjeta de evaluación ZC706 como SoC utilizado [12].

La interacción entre ambas partes se realiza mediante buses específicos, como AXI. Existen varios tipos de buses AXI en función de los requerimientos de la aplicación, pero en este caso se utiliza el AXI-Lite que se basa en la creación de un mapa de memoria típico. En la figura 2.6 se puede ver el procedimiento de lectura y escritura del bus.

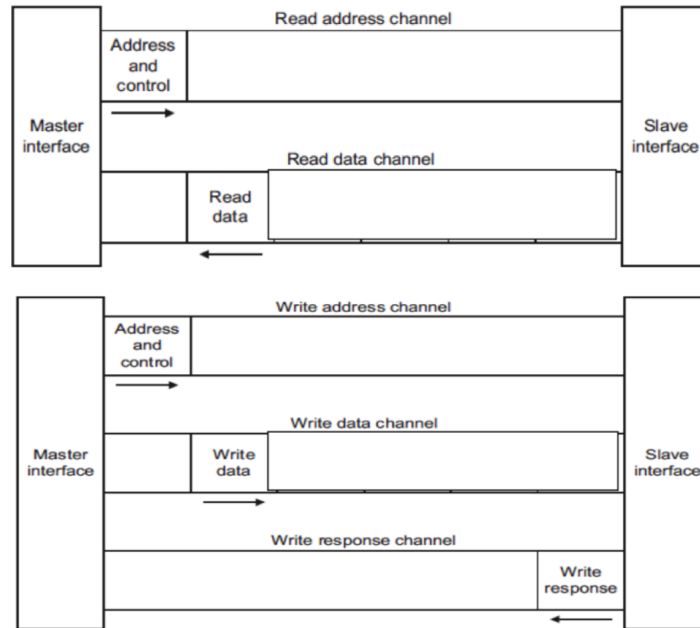
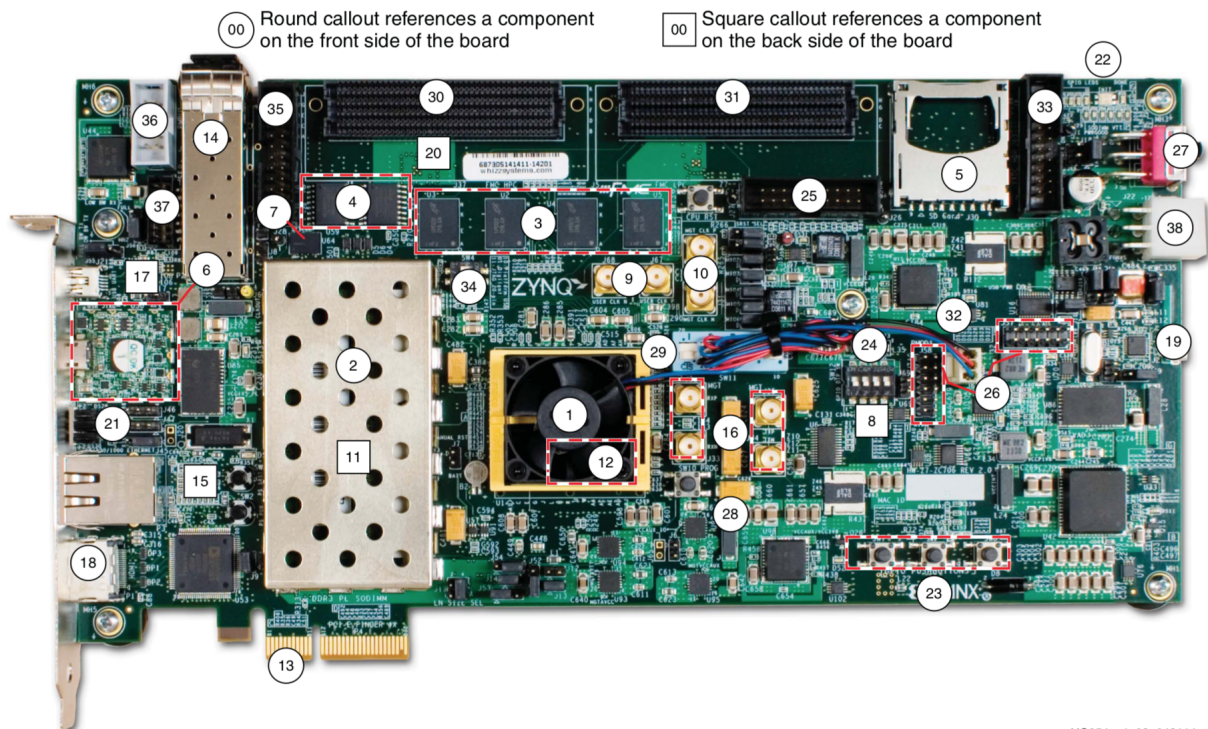


Figura 2.6: Lectura y escritura mediante el bus AXI-Lite [13].

La tarjeta utilizada es la Zynq-7000 SoC ZC706. Cuenta con dos procesadores Cortex-A9, en donde en uno se cargará una aplicación que realice determinadas tareas para ganar flexibilidad en el diseño. Después dispone de recursos lógicos suficientes para acometer el diseño. El conector del AFE sigue la especificación es FPGA Mezzanine Card (FMC), el SoC utilizado cuenta con dicho conector. En la figura 2.7 se puede ver la tarjeta utilizada, donde se enumeran todos los elementos que la componen y que vienen descritos en [14].



UG954\_c1\_02\_042114

Figura 2.7: Tarjeta de evaluación ZC706 utilizada para el desarrollo del trabajo [14].

## 2.3 Sistemas SoC en comunicaciones PLC

Se han desarrollado previamente otros trabajos en el campo de las comunicaciones PLC utilizando como dispositivo de desarrollo los sistemas SoC. A continuación se presentan varios proyectos dedicados a las comunicaciones PLC y las diferentes partes que abordan cada uno dentro del sistema de comunicación propio. Estos trabajos serán el punto de partida para la realización del presentado en este documento.

En [15][16] se encuentran trabajos que abordan diseños para la implementación de la técnica de acceso al medio que recomienda el estándar IEEE 1901-2010 para comunicaciones PLC de banda ancha, como es el banco de filtros multi-portadora (FBMC). FBMC es una técnica de acceso al medio basada en OFDM, a la que se le añade un banco de filtros para reducir la interferencia entre subportadoras. Entre otras ventajas, esto provoca que ya no sean necesarios los prefijos cíclicos en las tramas transmitidas, aumentando así la eficiencia espectral y robustez para mejorar la sincronización y compensar el mal comportamiento del canal PLC.

La arquitectura que proponen estos trabajos incluye un banco de filtros tras realizar la Discrete Fourier Transform (DCT), para provocar la mejora espectral comentada disminuyendo el Inter-Carrier Interference (ICI). Este banco de filtros se aplica a cada una de las subportadoras de la trama; cada trama cuenta con  $M = 512$  muestras/subportadoras. En la figura 2.8 se puede ver el esquema del diseño para implementar la técnica de acceso al medio. Existe un primer bloque de multiplicadores para realizar la multiplicación entre la entrada y unas constantes.

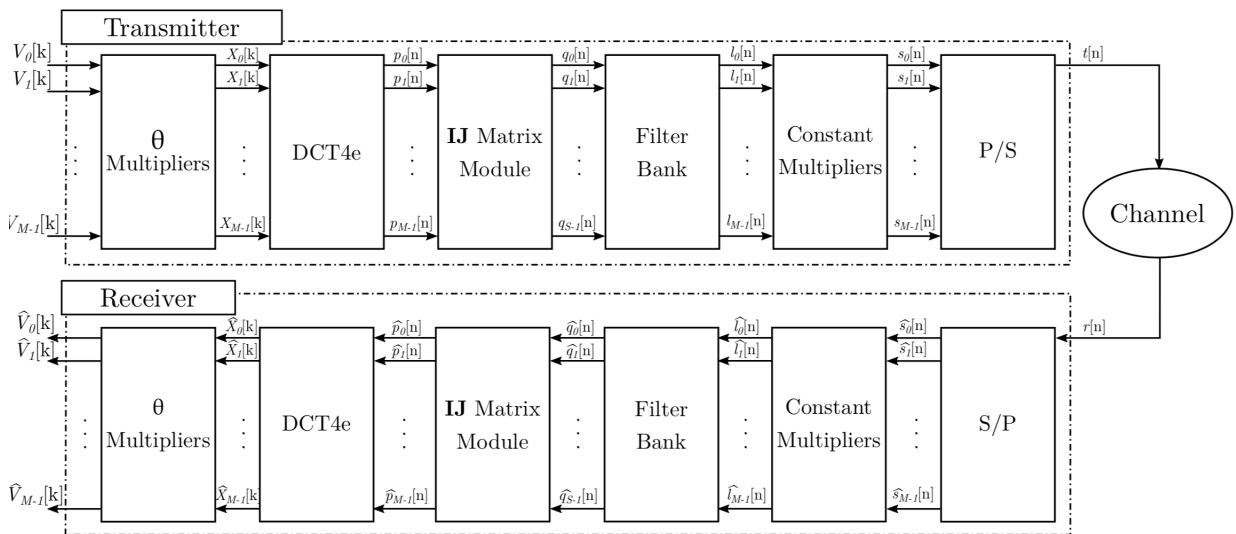


Figura 2.8: Diagrama de bloques del transmultiplexor FBMC [15].

Después de este primer bloque de multiplicadores, se encuentra la DCT-4e, cuya arqui-

itectura interna se representa en la figura 2.9. Esta transformada cuenta con un bloque inicial de multiplicadores, después se realiza una Fast Fourier Transform (FFT) de  $M$  puntos coincidiendo con el número de subportadoras por trama. Después se realiza una ordenación de los datos de salida de la FFT por la arquitectura que implementa la misma. Por último otro bloque de multiplicadores. Tras realizar la DCT con todos los bloques comentados que la componen, se realiza la multiplicación por las matrices  $IJ$ , que son la matriz unitaria y la matriz unitaria invertida; después se introduce en el banco de filtros que da nombre a esta modulación multi-portadora para mejorar la eficiencia espectral y reducir el ICI; a continuación, se introducen las subportadoras en un bloque de multiplicadores por una misma constante. Por último se serializan todas las subportadoras en un único bus. El transmisor y el receptor son iguales y presentan la misma arquitectura, pero se realizan de manera inversa, en recepción se debe paralelizar las subportadoras de las tramas que vienen serializadas por el transmisor.

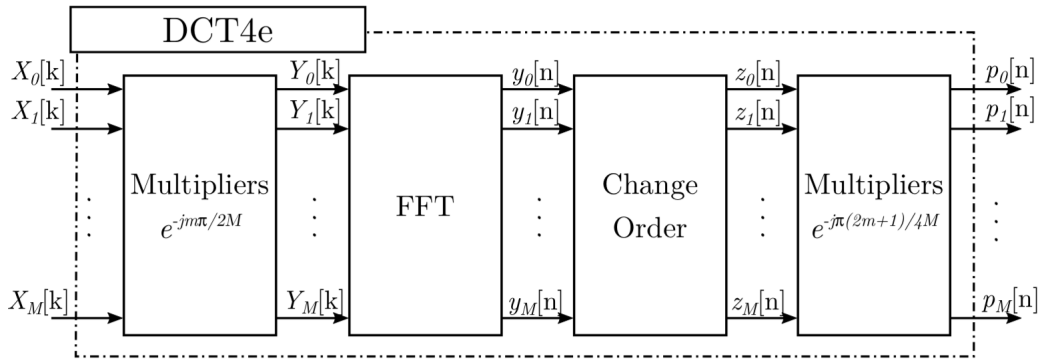


Figura 2.9: Diagrama de bloques de la DCT4e [15].

En [17] se encuentra el estudio del sincronismo, estimación e igualación de canal en comunicaciones PLC. Se prestará especial atención al método de sincronismo propuesto, donde se especifican las tramas de sincronismo más idóneas para canales PLC junto con su sincronismo. Se busca realizar la correlación cruzada con Complementary Set of Sequences (CSS) o también con secuencias Zadoff-Chu en el dominio de la frecuencia mediante una multiplicación. Para ello, es necesario realizar la FFT de las tramas recibidas y de las que se tienen de referencia para realizar la comparación y determinar cuándo dichas tramas son recibidas. Cuando se encuentren estas tramas, provocará un pico de correlación en el dominio temporal para determinar el inicio de la comunicación. Se contemplará este enfoque a la hora de realizar la arquitectura.

## Capítulo 3

# Diseño Propuesto

### 3.1 Introducción

Este capítulo tiene como finalidad explicar todo el proyecto y dar sentido a los capítulos posteriores, en los que se explicará en detalle cada parte del diseño desarrollado y su arquitectura correspondiente. Se realizará inicialmente una enumeración general de todo el diseño, y después, al tratarse de un sistema de comunicaciones, se procederá a realizar una descripción desde el comienzo de la transmisión hasta que se produce la recepción, comentando toda la arquitectura cronológicamente en el sentido de la misma. De esta manera, se puede ir entendiendo la necesidad de todos los módulos de los que dispone el diseño, y más adelante, se conocerá su configuración de manera más extensa y precisa en sus capítulos de descripción.

La figura 3.1 que se muestra a continuación es una ilustración que representa un SoC, descrito ya en capítulos anteriores, y en el que se puede diferenciar la existencia de una parte hardware o también denominada lógica programable o parte lógica (PL), que sería la FPGA; y otra parte denominada software o sistema de procesamiento (PS) que correspondería con el procesador ARM que lleva incorporado el SoC. En dicha figura se puede apreciar cómo se ha desglosado el diseño en ambas partes, aunque se explicarán a continuación y en posteriores capítulos en profundidad. También se debe apreciar en la figura 3.1, que existen dos dominios de reloj diferenciados en el diseño hardware. La parte izquierda de la imagen desde el *AFE Driver*, posee un dominio de reloj de 100 MHz, y la parte de la derecha uno de 50 MHz, correspondiendo con la frecuencia de funcionamiento del AFE físico. La arquitectura de los drivers del AFE servirá para realizar este cambio de dominio de reloj y manejar el funcionamiento del mismo. Se explicará cómo se realiza dicho cambio de dominio de reloj mediante memorias en el capítulo 5, dedicado al módulo *AFE Driver* en el que se describirá todo lo que realiza esta arquitectura y su composición.

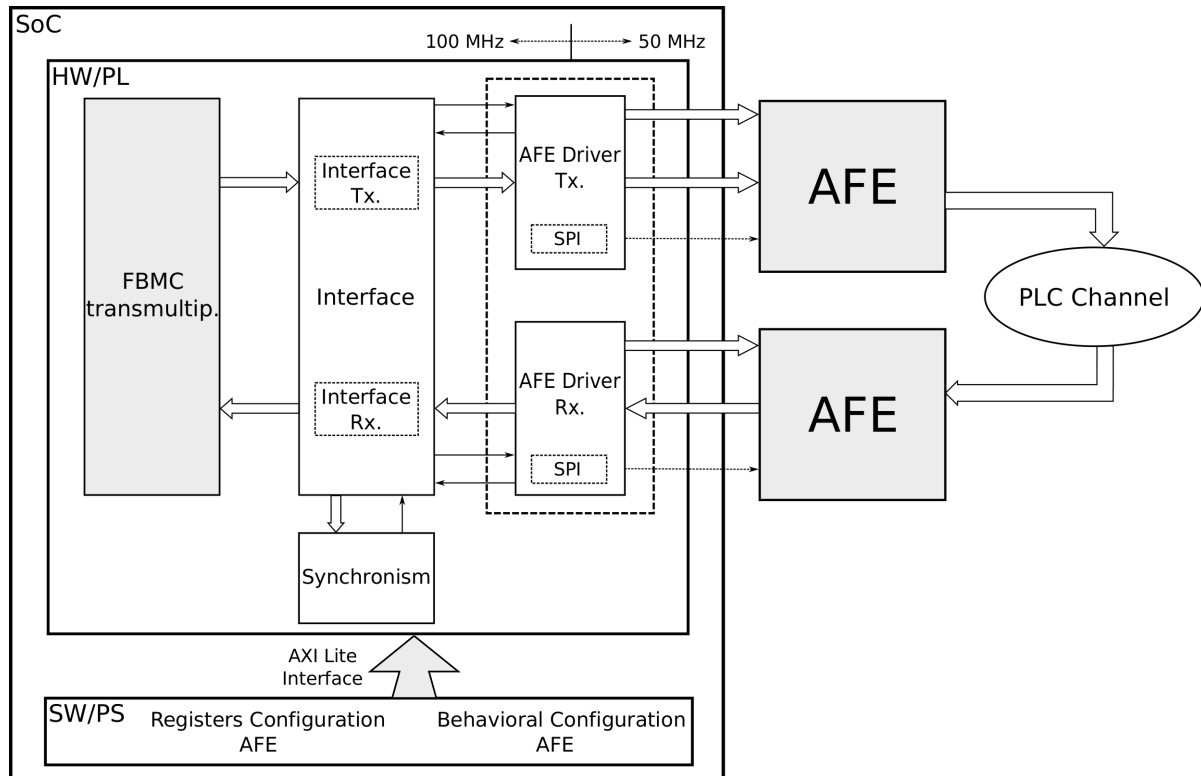


Figura 3.1: Esquema del diseño propuesto implementado en un SoC.

Para que el diseño disponga de varios relojes, es necesario añadir el IP CORE denominado Mixed Mode Clock Manager (MMCM). Al tratarse de un diseño complejo, se debe minimizar el retardo que sufren los relojes al distribuirse por la FPGA para que todas las arquitecturas estén sincronizadas; con este módulo se soluciona ese problema al repartir el reloj por las pistas destinadas al mismo. Dicho bloque genera múltiples relojes en función del reloj que se introduce a su entrada, en la cual, se establecerá un reloj de la ZYNQ configurado a 100 MHz, y se obtendrán 2 salidas, una señal de reloj de 50 MHz y otra de 50 MHz desfasada 180°.

La interacción entre ambas partes (parte lógica PL y parte software PS) se realizará mediante el protocolo AXI (ya comentado en la introducción como una de las herramientas que se establecen en el codiseño) y que implementa ARM, mediante su bus de comunicaciones, en este caso con la especificación AXI Lite, con sus cronogramas correspondientes. A través del IP Interface (IPIF) se puede obtener el valor de los registros escritos en software y utilizarlos para configurar y establecer la arquitectura del hardware que se requiera. Todas las arquitecturas que requieran obtener algún valor establecido en la parte software, deberá implementar un IPIF en su arquitectura. En este diseño lógicamente al utilizar ambas partes del SoC, habrá varias arquitecturas que lo necesiten.

## 3.2 Descripción general del diseño

En la parte hardware, denominada en la figura anterior como Hardware (HW), se puede encontrar el diseño de la arquitectura planteado con todos los bloques que harán posible el funcionamiento del sistema. Las arquitecturas que componen el diseño son las siguientes:

- Transmultiplexor FBMC:
  - Dicho bloque representa la arquitectura que implementa la técnica de acceso al medio FBMC.
  - Consta tanto de la arquitectura que realiza la transmisión, como de la que realiza la recepción. Para hacer más sencilla la figura en la que se aprecia el diseño completo, se representa como un único bloque. La arquitectura del transmisor y del receptor es la misma, y el diagrama de bloques ya fue explicado en el capítulo 2, por lo que no se volverá a profundizar en este módulo del diseño.
- Interfaz:
  - Esta arquitectura realiza un papel vital para el funcionamiento de todo el sistema. Como su propio nombre indica, sirve de interfaz entre todas las arquitecturas del sistema y realiza las transformaciones oportunas para la correcta comunicación de todos los bloques.
  - Se encarga de adaptar la transmisión y recepción de la técnica de acceso al medio FBMC a las limitaciones físicas del AFE:
    - \* Realiza una cuantificación del tamaño de palabra de cada subportadora, al tamaño del bus de datos bidireccional del AFE (10 bits).
    - \* Realiza en el bloque transmisor una conversión de complemento a-2 a binario natural para que los conversores del AFE realicen bien su función.
    - \* Realiza en el bloque receptor una conversión de binario natural a complemento a-2 para devolver a los datos el formato original que requiere el receptor FBMC.
    - \* Realiza una cuantificación al tamaño original de las subportadoras (24 bits), estableciendo los 10 bits del bus del AFE como los de mayor peso.
  - Gestiona todos los procesos que se realizan en el dominio de reloj de 100 MHz de la arquitectura *AFE Driver*, en función de los flags que reciba de las memorias First-Input First-Output (FIFO) de dichos drivers. Se implementa una lógica de control mediante la lectura de estos flags para poder tener transmisiones con anchos de banda superiores del máximo que permite el AFE.
  - Ofrece soporte a la arquitectura encargada del sincronismo del diseño que se sitúa en el bloque receptor del diseño:



- \* En primer lugar, en la parte transmisora de la interfaz, almacena las tramas de sincronismo y las tramas de información que se reciben mientras se transmiten estas primeras, para así de esta manera, no perder ninguna subportadora mientras se transmiten las de sincronismo previamente.
  - \* En segundo lugar, en la parte receptora de la interfaz, almacena todas las muestras recibidas del AFE receptor mientras se decide dónde se encuentra la primera subportadora de la primera trama mediante el módulo de *Sincronismo*, para una vez determinada la posición, obtener todas las tramas sincronizadas para transmitirle al receptor FBMC.
  - En el capítulo 7 se realizará una descripción en detalle de este módulo.
  - Driver del AFE Transmisor
    - Arquitectura *AFE Driver* configurada como transmisora vía software.
    - Realiza la configuración física del AFE mediante la transmisión de señales para establecer en sus pines.
    - Se encarga del cambio de dominio de reloj (de 100 MHz a 50 MHz).
    - Selección de las características de funcionamiento del AFE mediante la configuración de registros a través del protocolo SPI.
    - Gestión de todos los procesos en el dominio de reloj de 50 MHz.
    - En el capítulo 5 se realiza una descripción en detalle de este módulo.
  - Driver del AFE Receptor
    - Arquitectura *AFE Driver* configurada como receptora vía software.
    - Realiza la configuración física del AFE mediante la transmisión de señales para establecer en sus pines.
    - Se encarga del cambio de dominio de reloj (de 50 MHz a 100 MHz).
    - Selección de las características de funcionamiento del AFE mediante la configuración de registros a través del bus SPI.
    - Gestión de todos los procesos en el dominio de reloj de 50 MHz.
    - En el capítulo 5 se realiza una descripción en detalle de este módulo.
- Ambas arquitecturas (AFE Driver TX y AFE Driver RX) son la misma, pero se han configurado vía software para que funcionen de manera diferente.
- Sincronismo
    - Arquitectura que busca sincronizar el transmisor y el receptor para poder así conocer dónde comienza la primera trama de información transmitida.



- Dicha arquitectura realizará la correlación cruzada en el dominio de la frecuencia mediante un producto. Esta operación se utiliza a la hora de realizar comparaciones, que es la finalidad de este módulo, reconocer las tramas de sincronismo enviadas para poder así determinar el inicio de la primera trama de datos.
- El capítulo 6 está dedicado a la descripción en detalle de este módulo.

En la parte Software (SW) denominada en la figura que representa todo el diseño, se encontrará la programación en un lenguaje de alto nivel para después configurar en función de ella la parte hardware. Lo que se define en software corresponde con los siguientes puntos.

- Señales a establecer en los pines del primer AFE que se configurará como transmisor.
- Señales a establecer en los pines del segundo AFE que se configurará como receptor.
- Registros del AFE configurado como transmisor para establecer las características deseadas.
- Registros del AFE configurado como receptor para establecer las características deseadas.
- Configuraciones y elementos necesarios para el funcionamiento de la arquitectura de la técnica de acceso al medio FBMC.

Mediante estas descripciones se pretende establecer un lugar de referencia para buscar de manera rápida algún elemento del diseño y poder acceder al capítulo adecuado para su explicación en detalle.

### 3.3 Descripción y configuración del diseño en el sentido de la comunicación

Este trabajo plantea un sistema de comunicación completo, por lo que tendrá un bloque transmisor y uno receptor. Como ya se comentó al principio del capítulo, se comenzará la descripción desde el transmisor para terminar en el receptor.

Inicialmente en SW, se programará la configuración de cada *AFE Driver*, con sus registros correspondientes, uno como transmisor y otro como receptor con las características correspondientes para transmitirle al AFE. Además, en cada arquitectura del driver, se dispone de un Serial Peripheral Interface (SPI) por el que transmitir la información al AFE, por lo que el valor de los registros con las características correspondientes, también se programará en software y se transmitirá a dicha arquitectura para que configure el AFE.

Estos valores que se establecen en los registros en la parte software pueden ser modificados en cualquier momento, y con una reprogramación del procesador del SoC se obtendría otro funcionamiento del HW sin la necesidad de implementar toda la arquitectura de nuevo. Es por lo que ofrece muchas ventajas establecer toda la configuración de la arquitectura en la parte SW. En la figura 3.2 se puede ver qué partes del hardware son configuradas o reconfiguradas a través del software mediante la programación del procesador.

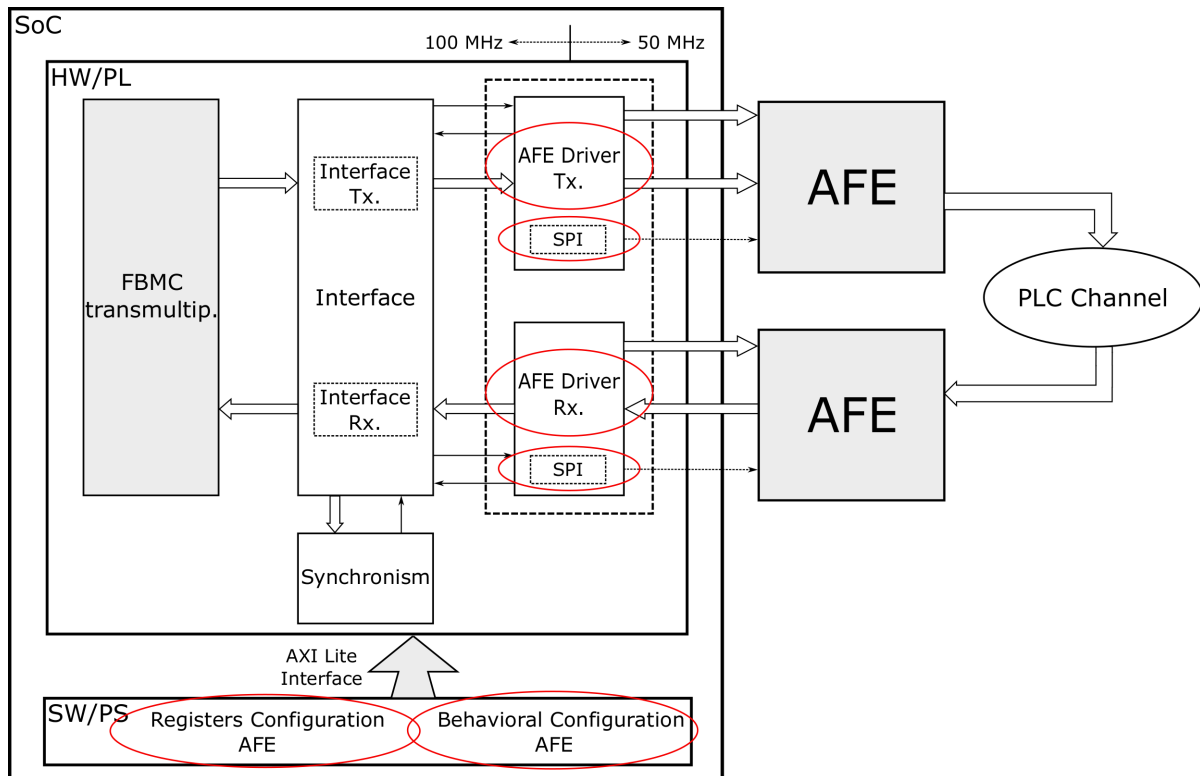


Figura 3.2: Interacción del software en el diseño hardware para establecer la configuración y funcionalidad del AFE correspondiente.

Se procede ahora con una descripción ordenada en el sentido de la comunicación de los elementos que componen el diseño hardware. La transmisión de la información comienza con la habilitación del FBMC transmisor, de donde se obtienen tramas de 512 subportadoras, a una frecuencia de 50 MHz. Como se comentó al principio del capítulo, esta parte del diseño funciona a una frecuencia de 100 MHz, por lo que mediante retardos después del envío de cada trama, se consigue obtener los 50 MHz. Aunque se ajusta el transmisor FBMC a 50 MHz para adaptarse a las limitaciones del AFE, para darle flexibilidad al diseño, en el módulo *Interfaz* se implementa una lógica de control basada en el estado del almacenamiento en el driver transmisor para poder tener transmisiones a mayor frecuencia, aunque conllevaría pérdida de información.

Después, se realiza una cuantificación de cada subportadora de la trama a un tamaño de palabra de 10 bits, para que concuerde con el tamaño del bus de datos bidireccional del

AFE. Además, se convierten de complemento a-2 a binario natural para que los conversores realicen una correcta interpretación de los datos.

Realizado todo el tratamiento de las subportadoras de la trama, se procede a sincronizar el transmisor con el receptor como en la mayoría de las comunicaciones. Para ello, se ha utilizado una memoria BRAM en la que se almacenarán dos tramas de sincronismo de 512 muestras cada una para mandar al bloque receptor a través del canal PLC. En paralelo a esta memoria BRAM, se añade una memoria FIFO para no perder las subportadoras que envía el transmisor FBMC mientras se transmiten las tramas de sincronismo. Una vez se han enviado las dos tramas de sincronismo con una longitud de 1024 muestras, se conmuta a la memoria FIFO para enviar las subportadoras almacenadas del transmisor y las futuras que reciba. En la descripción del bloque receptor se explicará que se hace con las tramas de sincronismo cuando se reciben.

Como ambas memorias (BRAM y FIFO) trabajan en el dominio de reloj de los 100 MHz, y el AFE funciona a 50 MHz, será necesario implementar una lógica adicional para sacar una muestra cada dos ciclos de reloj de 100 MHz. Esto facilitará al *AFE Driver* transmisor el cambio de dominio sin perder ninguna muestra transmitida. En la figura 3.3 se puede ver lo que se ha comentado en los párrafos anteriores de una forma más visual.

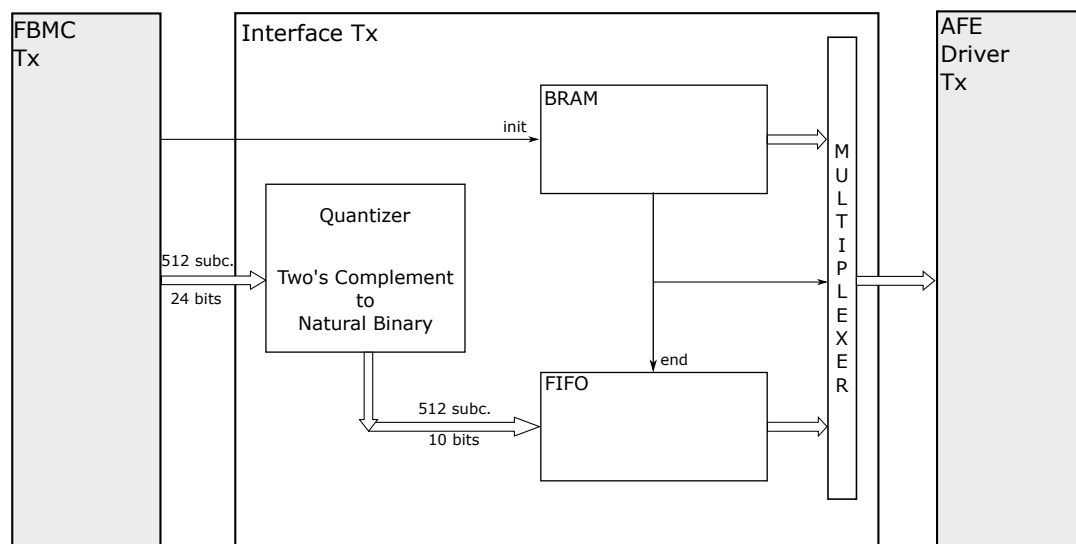


Figura 3.3: Transmisión inicial de las tramas de sincronismo y conmutación de memorias al iniciar la comunicación.

Tanto las tramas de sincronismo, como las subportadoras que envía el transmisor FBMC, salen de la arquitectura *Interfaz* para dirigirse a la arquitectura *AFE Driver* configurada como transmisor. La función de esta arquitectura consiste en configurar el AFE de la manera establecida en el SW y realizar el correcto cambio de dominio de reloj debido a que el sistema funciona a una frecuencia superior con respecto a la que trabaja AFE (50 MHz).

Para realizar estas dos principales tareas, el *AFE Driver* evalúa los parámetros establecidos por el SW y determina si se va a configurar como transmisor o receptor, aplicando las señales correspondientes para transmitir a los pines del AFE. Los valores establecidos en el software se obtienen de la arquitectura del IPIF, que actúa de interfaz entre la parte lógica y la software. Para establecer el cambio de dominio de reloj, se colocan dos FIFO (una para si se configura en modo transmisión y otra por si se configura en modo recepción) en las que ir almacenando las muestras que se transmitirán posteriormente al AFE. El control de escritura en la memoria FIFO de transmisión o el control de lectura en la memoria FIFO de recepción se realiza en la arquitectura *Interfaz*; la lectura en transmisión y escritura en recepción la realizará la arquitectura del *AFE Driver*, que se realiza a 50 MHz. A la salida de la memoria FIFO del driver transmisor existe un registro tri-estado, esto se debe a que el bus de datos del AFE es bidireccional, y la arquitectura del driver puede trabajar tanto de transmisora como de receptora. El registro tri-estado pondrá datos en el bus bidireccional en caso de estar transmitiendo, o pondrá alta impedancia en esa salida para poder recibir los datos del canal PLC si trabaja como receptor. Además, dicha arquitectura dispone de un módulo SPI, con el cual configura los registros del AFE físico para establecer las características deseadas. En la figura 3.4 se puede ver lo que ha descrito en este párrafo.

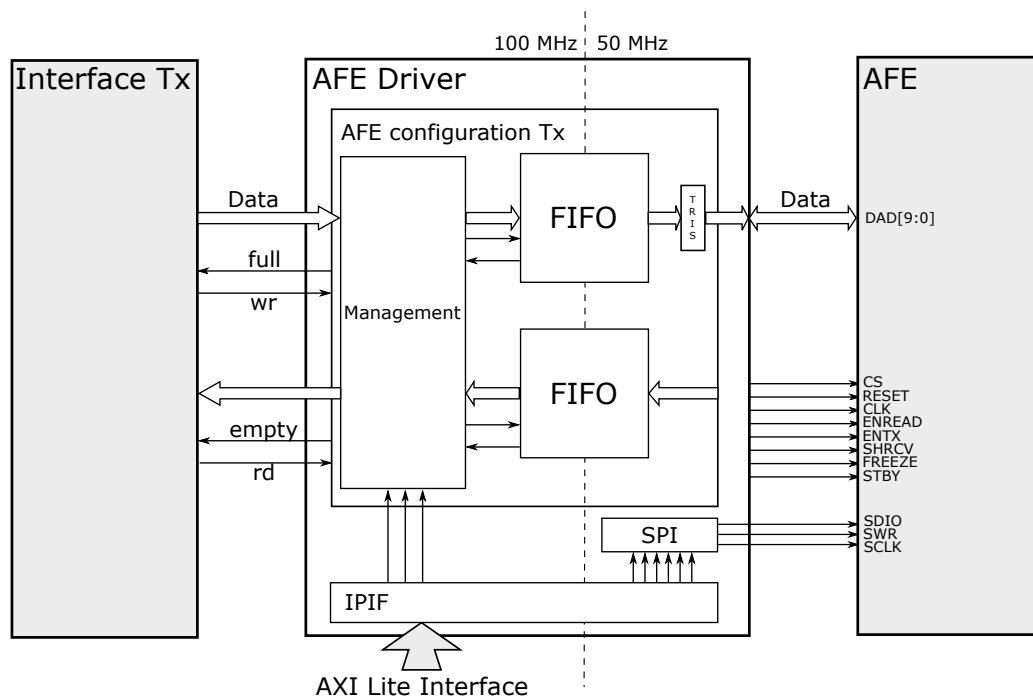


Figura 3.4: Arquitectura del driver del AFE.

Del *AFE Driver* salen los datos hacia el AFE transmisor, que al haberlo configurado en este modo realizará la conversión a través del Digital-Analog Converter (DAC) para poder transmitirlos a través de la red eléctrica a la que se accede mediante un enchufe convencional conectado al propio AFE. La transmisión de la información se realiza a 50

Msp/s, como cada muestra tiene 10 bits, se realiza una transmisión a una tasa binaria de 500 Mbps.

Una vez llegan los datos al otro AFE a través del canal PLC, debido a que se ha configurado en modo recepción el hardware como se definió en el software, se utilizará el Analog-Digital Converter (ADC) para convertir la señal analógica que recibe por el canal en una señal digital que puede ser tratada por el *AFE Driver* configurado para recepción. Como ya se comentó anteriormente, el cambio de dominio de reloj se realiza a través de una memoria FIFO, pero en este caso, de 50 MHz a 100 MHz. Como la frecuencia de funcionamiento del resto del sistema es superior a la de salida de datos del AFE por parte del ADC, no será necesario una memoria FIFO de gran tamaño, ya que en cuanto se escriban van a ser leídas siguiendo la lógica implementada en el módulo *Interfaz*. Se profundizará en esta descripción en el capítulo 7.

Una vez que las muestras pasan por el *AFE Driver* y se transmiten hacia el receptor FBMC, lo hacen en dos direcciones. Hacia el módulo *Interfaz* para almacenarse en una memoria FIFO, donde se guardan los datos mientras se realiza el proceso de sincronismo, y hacia 4 memorias DUAL PORT para poder darle al módulo de *Sincronismo* dos bloques de 512 valores y poder detectar el pico de correlación cuando se traten las tramas de sincronismo enviadas. La memoria FIFO alojada en la interfaz será de un gran tamaño, ya que no puede perder ninguna trama mientras se realiza todo el proceso de sincronismo, que consume una cantidad elevada de ciclos en latencia. En la figura 3.5 se puede ver la bifurcación y cómo se realizarán los siguientes pasos.

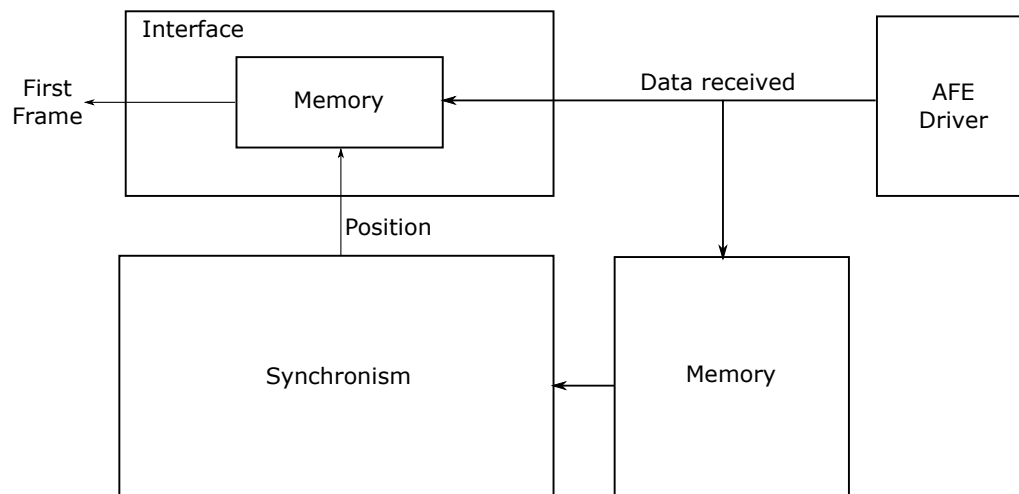


Figura 3.5: Bifurcación de la información recibida del canal PLC para realizar el sincronismo.

Toda la arquitectura del sincronismo es realizada mediante la herramienta de síntesis de alto nivel de Xilinx llamada HLS. Mediante la programación en un lenguaje de alto nivel como es C++, se consigue la arquitectura hardware. Ya se explicaron las ventajas y los

inconvenientes de esta herramienta en la introducción.

Para comprobar el funcionamiento de este sincronismo, se realiza un modelo en Matlab con representación en coma fija que simule dicho bloque. Una vez se verifica que funciona correctamente, como es el caso, se implementa. El bloque de sincronismo cuenta con una primera FFT a las tramas de sincronismo recibidas del canal PLC, después se realiza una multiplicación entre el resultado de la primera FFT y la FFT del conjugado de las tramas de sincronismo ideales, que son almacenadas previamente en una memoria y con las que se realiza la comparación. Realizada la correlación en el dominio de la frecuencia, se debe realizar la IFFT del resultado para volver al dominio del tiempo y obtener el pico de correlación; dicha arquitectura de la FFT es la misma que la realizada en el trabajo fin de grado, pero mediante la síntesis de alto nivel de HLS. Para finalizar, se busca la posición del pico de correlación y se determina dónde se encuentra el comienzo de la primera trama de datos transmitida. Una vez se obtiene la posición, se busca la primera subportadora en la memoria FIFO que almacenaba en paralelo al proceso de sincronismo todas las muestras recibidas. En la figura 3.6 se puede ver esta parte del proceso para conseguir sincronizar el transmisor y el receptor.

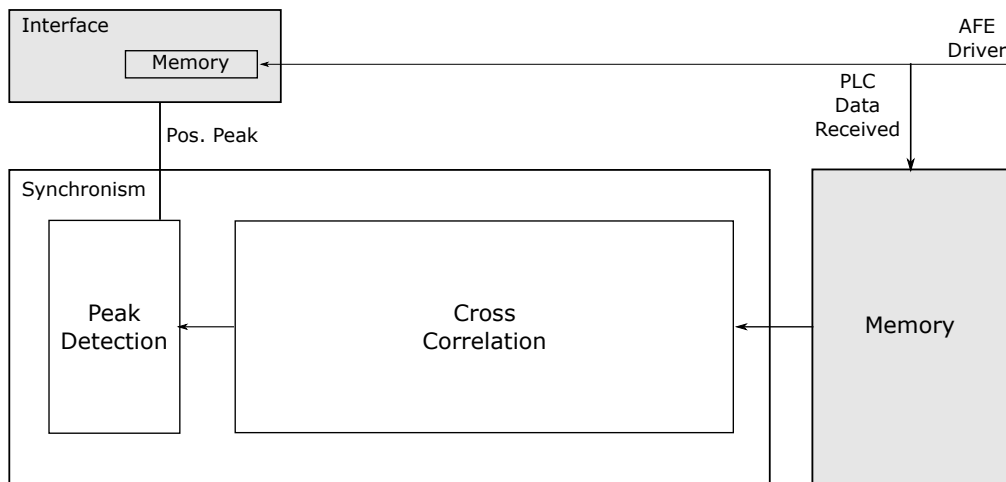


Figura 3.6: Proceso de sincronismo llevado a cabo en el diseño.

Obtenidas la primera subportadora de la trama, se almacenan las 512 subportadoras que la componen en orden en una memoria *ping-pong*, para que se trasmiten de manera inversa por requerimientos del FBMC receptor y a una frecuencia de 100 MHz al receptor FBMC, y poder de esta manera obtener la información generada en primera instancia.

## Capítulo 4

# Descripción del módulo AFE (Analog Front-End)

### 4.1 Introducción

En este capítulo se describe un elemento primordial para llevar a cabo la propuesta del trabajo, el AFE. Se trata de un dispositivo que se emplea, principalmente y entre otras funciones, para realizar la conversión digital-analógica en el bloque transmisor de la arquitectura, y la analógica-digital en el receptor. El módulo AFE empleado en este trabajo cuenta con ambos conversores DA/AD, pero solo tiene un único bus de datos bidireccional de 10 bits, por lo que será necesario realizar un driver en donde se establezca uno de los dos tipos de configuración: transmisora o receptora. De esta manera en las pruebas preliminares, será necesario contar con dos AFE en el diseño, uno se configura como transmisor para realizar la conversión digital-analógica y enviar tramas de datos al canal PLC; y otro configurado como receptor para realizar la conversión analógica-digital y enviar al bloque receptor del diseño las tramas recibidas desde el canal. El AFE tiene una frecuencia de conversión de 50 MHz con un tamaño de palabra del bus de 10 bits, por lo que el dispositivo funciona a una tasa de 50 Msps. Se obtienen por tanto comunicaciones con un ancho de banda de 25 MHz. Este ancho de banda supone no cumplir las especificaciones del estándar IEEE 1901-2010 para comunicaciones PLC de banda ancha, que requieren un ancho de banda de 31.25 MHz. La arquitectura realizada puede trabajar cumpliendo con el ancho de banda establecido, pero por limitaciones del AFE, se hace con el ancho de banda máximo que permite, los 25 MHz.

El AFE no se utiliza en exclusiva como elemento conversor, también cuenta con etapas de amplificación, acoplamiento y aislamiento. En la figura 4.1 se puede ver una representación simplificada del MAX2981 de Maxim Integrated, elemento central del módulo AFE empleado. Se puede ver que, en la parte transmisora, cuenta con un DAC y un filtro paso bajo para eliminar los armónicos y los espúreos adyacentes a la banda de paso deseada;

en la parte receptora, se puede ver en la parte inicial dos filtros (paso alto y paso bajo) para eliminar ruidos no deseados y evitar aliasing para un muestreo preciso del ADC, un AGC para escalar la señal y el propio ADC.

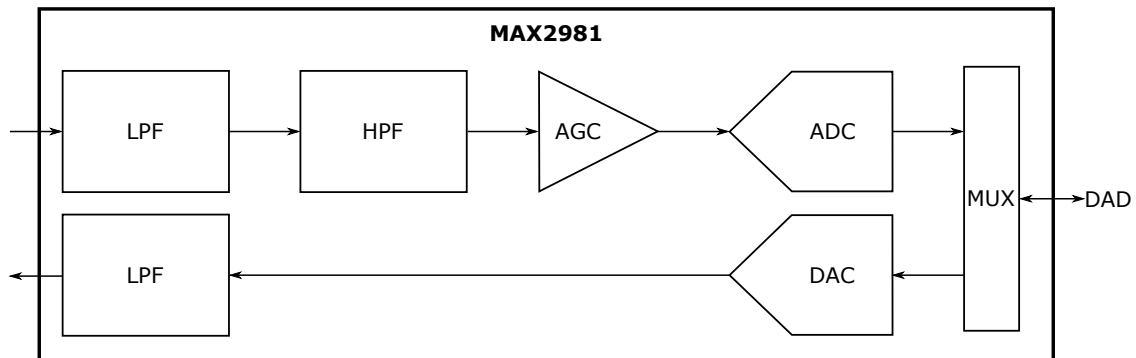


Figura 4.1: Diagrama de bloques simplificado del funcionamiento del dispositivo MAX2981 [18].

En la figura 4.2 se muestra el AFE físico utilizado para la propuesta, conectado a la tarjeta de evaluación ZC706 (basada en una FPGA) mediante un conector FMC, y donde se aprecia el MAX2981 y el transformador encargado de acoplar la señal analógica convertida a los 220 V de la red eléctrica.

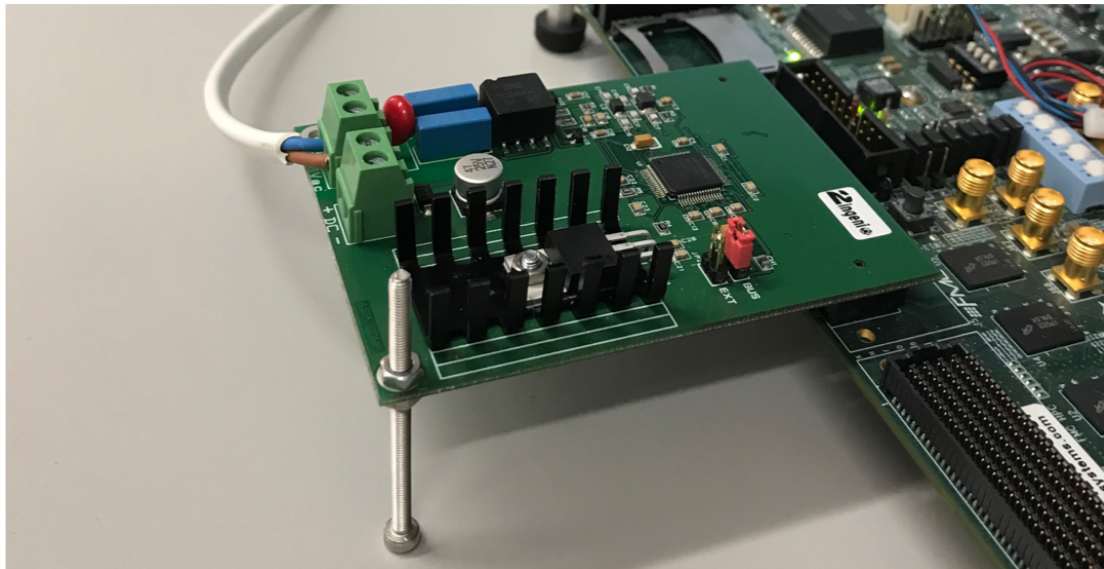


Figura 4.2: Montaje del AFE en la tarjeta mediante su conector FMC.

El AFE cuenta con registros internos con los cuales poder modificar sus características en función del valor que se establezca. Para modificar el valor de los mismos, establece líneas serie para la transmisión de los nuevos parámetros. Será necesario realizar un protocolo SPI en la arquitectura del driver con el que transmitir el valor de los registros bit a bit.

El AFE reduce de manera considerable el número de componentes necesarios para realizar la transmisión y recepción al/del canal PLC. Esto provoca que el AFE sea una opción económica y flexible para afrontar este tipo de comunicaciones.





Tabla 4.1: Descripción de los pines del AFE.

Descripción de pines		
Pin	Nombre	Función
1, 5, 9, 10, 13, 17, 28, 32, 52, 53, 56, 57	AGND	Masa analógica.
2, 6, 12, 15, 16, 29, 54, 55, 60	AVDD	Voltaje suplementario de potencia analógica.
3	PLIP	Entrada positiva de la red eléctrica AC.
4	PLIN	Entrada negativa de la red eléctrica AC.
7	CEXT	Conexión del condensador externo.
8	REXT	Conexión de la resistencia externa.
11	PLOP	Salida positiva de la red eléctrica AC.
14	PLON	Salida negativa de la red eléctrica AC.
18	REGOUT	Salida reguladora de voltaje.
19, 26, 49	DVDD	Entrada de voltaje digital 2.4 V.
20, 27, 34, 40, 47, 50	DGND	Masa digital.
21	SDIO	Entrada de datos serie.
22	SCLK	Entrada del reloj serie.
23	SHRCV	Control de apagado del receptor.
24	ENREAD	Control de la habilitación del modo lectura.
25	CS	Entrada de selección de portadora.
30, 37, 41, 44	DVDD3	Voltaje suplementario digital.
31	CLK	Entrada del reloj del sistema (50 MHz).
33	DAD9	Bit de mayor peso de la entrada/salida del DAC/ADC.
35	DAD8	Bit 8 de la entrada/salida del DAC/ADC.
36	DAD7	Bit 7 de la entrada/salida del DAC/ADC.
38	DAD6	Bit 6 de la entrada/salida del DAC/ADC.
39	DAD5	Bit 5 de la entrada/salida del DAC/ADC.
42	DAD4	Bit 4 de la entrada/salida del DAC/ADC.
43	DAD3	Bit 3 de la entrada/salida del DAC/ADC.
45	DAD2	Bit 2 de la entrada/salida del DAC/ADC.
46	DAD1	Bit 1 de la entrada/salida del DAC/ADC.
48	DAD0	Bit de menor peso de la entrada/salida del DAC/ADC.
51	FREEZE	Establece el modo congelado en el receptor.
58, 59	I.C.	Conectado internamente.
61	ENTX	Habilitación de la transmisión.
62	SWR	Escritura de los registros.
63	RESET	Entrada de reset.
64	STBY	Entrada de standby.

4.4 se pueden ver las señales que se utilizan del AFE y la etiqueta que se asigna a la hora de buscar el pin correspondiente en el HPC o LPC.

En la tabla 4.2 se establece la relación entre las señales del AFE utilizadas y los pines de los dos conectores FMC.

#### 4.2.2 Restricciones temporales

Todo diseño requiere hacer un estudio del sistema para determinar si es necesario establecer una serie de restricciones temporales. Estas restricciones permiten ajustarse a los parámetros físicos y poder así funcionar de manera adecuada. Es el caso de este dise-

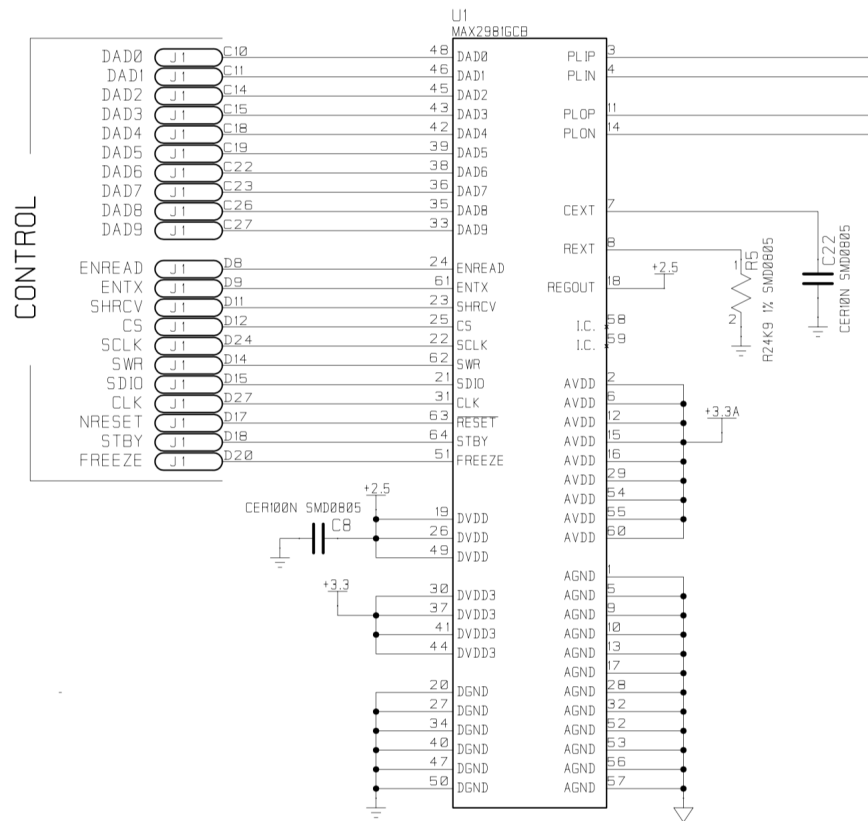


Figura 4.4: Pines del MAX2981 utilizado como AFE.

Tabla 4.2: Terminales de los conectores FMC para conectar a los puertos del diseño de la arquitectura.

Asignación de pines del conector a los puertos			
Pines AFE	Pin HPC/LPC J5	Pin FPGA HPC	Pin FPGA LPC
DAD0	C10	AG22	AB12
DAD1	C11	AH22	AC12
DAD2	C14	AG24	AC14
DAD3	C15	AG25	AC13
DAD4	C18	AC24	AF18
DAD5	C19	AD24	AF17
DAD6	C22	W25	AE27
DAD7	C23	W26	AF27
DAD8	C26	V28	AJ28
DAD9	C27	V29	AJ29
ENREAD	D8	AG21	AF15
ENTX	D9	AH21	AG15
SHRCV	D11	AH23	AE16
CS	D12	AH24	AE15
SCLK	D24	P26	AK26
SWR	D14	AD21	AH14
SDIO	D15	AE21	AH13
CLK	D27	T28	AK30
RESET	D17	AA22	AH17
STBY	D18	AA23	AH16
FREEZE	D20	V23	AB27

ño, en el cual se tiene un sistema funcionando en flanco de subida, y el AFE tiene unos conversores que funcionan en flanco de bajada. Estos conversores tienen un tiempo de establecimiento del dato, en el que el mismo debe de permanecer constante. El problema de este diseño radica en el bloque transmisor y el DAC del AFE, el sistema transmite datos en cada flanco de subida; es por tanto necesario aplicar una restricción temporal que lo enganche con el flanco de bajada posterior. Dado que trabaja a 50 MHz, cada ciclo de reloj suponen 20 ns, por lo tanto, como máximo debe retrasarse 10 ns, pero por asegurar que el AFE toma el valor cumpliendo el tiempo de establecimiento, se establecen 8 ns y así se asegura la captura. En la figura 4.5 se puede ver el cronograma de conversión del DAC comentado.

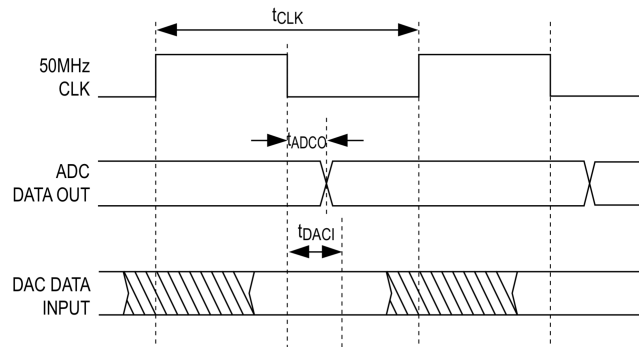


Figura 4.5: Cronograma del tiempo de mantenimiento del ADC/DAC que incorpora el AFE [18].

### 4.3 Registros internos del AFE

El AFE cuenta con 6 registros internos que determinan sus características de funcionamiento. De estos 6 registros, solo se pueden configurar 4, y no todos los bits del registro se utilizan para establecer una función concreta. En la figura 4.6 se puede ver cómo se deben escribir dichos registros. Inicialmente, se debe mandar el valor del registro, empezando por el bit de menor peso, y concatenando el valor del registro por su identificador empezando por el bit de menor peso también y especificado en la tabla 4.3.

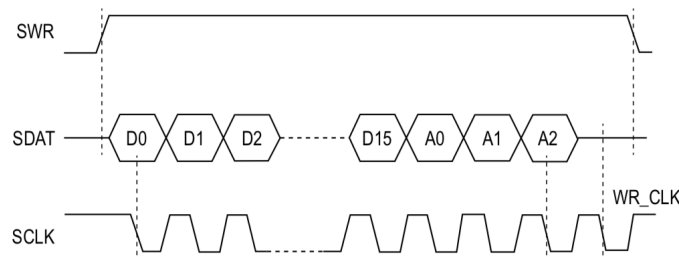


Figura 4.6: Descripción de la escritura de registros internos del AFE dada por el fabricante Maxim Integrated [18].

Tabla 4.3: Direcciones de los registros del AFE.

Direcciones			
Registro	A2	A1	A0
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1

En la tabla 4.4 se representa el registro número 1; como se puede observar, solo se pueden modificar los cuatro primeros bits. Este registro se encarga de habilitar y deshabilitar tanto la parte transmisora como receptora, o el dispositivo completo.

Tabla 4.4: Configuración del registro 1 del AFE.

Registro N°1		
Bit	Valor por defecto	Descripción
0	Nivel alto	A nivel alto, apaga el receptor cuando está en modo transmisor.
1	Nivel alto	A nivel alto, apaga el transmisor cuando está en modo receptor.
2	Nivel bajo	A nivel alto, apaga el DAC cuando está en modo receptor.
3	Nivel bajo	A nivel alto, apaga el dispositivo entero.
4	Nivel bajo	Reservado.
5	Nivel bajo	Reservado.
6	Nivel bajo	Reservado.
7	Nivel bajo	Reservado.
8	Nivel bajo	Reservado.
9	Nivel bajo	Reservado.
10	Nivel bajo	Reservado.
11	Nivel bajo	Reservado.
12	Nivel bajo	Reservado.
13	Nivel bajo	Reservado.
14	Nivel bajo	Reservado.
15	Nivel bajo	Reservado.

En la tabla 4.5 se representa el registro número 2; como se puede observar, solo se puede modificar el bit de mayor peso, con el cual se habilita o salta el filtro paso bajo de recepción.

En la tabla 4.6 se representa el registro número 3. Como se puede observar, solo se pueden configurar cuatro bits. Con tres de ellos se establece la ganancia del AGC de la configuración receptora del AFE, llegando a doblar la señal con 3dB extra o dividirla entre 4 con -6dB. El otro bit configura la adaptación de este AGC.

En la tabla 4.7 se representa el registro número 6. Como se puede observar, solo se pueden modificar cuatro bits. Con ellos, se pueden cambiar la configuración del Line Driver (LD), la habilitación o deshabilitación del filtro paso bajo del bloque transmisor del AFE, o del filtro paso alto del bloque receptor.

Tabla 4.5: Configuración del registro 2 del AFE.

Registro N°2		
Bit	Valor por defecto	Descripción
0	Nivel bajo	Reservado.
1	Nivel bajo	Reservado.
2	Nivel bajo	Reservado.
3	Nivel alto	Reservado.
4	Nivel bajo	Reservado.
5	Nivel bajo	Reservado.
6	Nivel bajo	Reservado.
7	Nivel bajo	Reservado.
8	Nivel bajo	Reservado.
9	Nivel bajo	Reservado.
10	Nivel bajo	Reservado.
11	Nivel bajo	Reservado.
12	Nivel bajo	Reservado.
13	Nivel bajo	Reservado.
14	Nivel bajo	Reservado.
15	Nivel bajo	A nivel alto, puentea el filtro paso bajo del receptor.

Tabla 4.6: Configuración del registro 3 del AFE.

Registro N°3		
Bit	Valor por defecto	Descripción
0	Nivel alto	Reservado.
1	Nivel alto	Reservado.
2, 3, 4	Nivel bajo	Establece la ganancia del AGC del receptor en el rango [000, 111]: 3dB, 2dB, 1dB, 0dB, -1dB, -2dB, -3dB, -6dB.
5	Nivel bajo	Reservado.
6	Nivel bajo	Reservado.
7	Nivel bajo	Reservado.
8	Nivel bajo	Reservado.
9	Nivel bajo	Reservado.
10	Nivel bajo	Reservado.
11	Nivel alto	A nivel alto, establece el proceso de ajuste en modo continuo. En caso contrario solo se activa durante el reset.
12	Nivel bajo	Reservado.
13	Nivel alto	Reservado.
14	Nivel alto	Reservado.
15	Nivel alto	Reservado.

Tabla 4.7: Configuración del registro 6 del AFE.

Registro N°6		
Bit	Valor por defecto	Descripción
0	Nivel bajo	Reservado.
1, 2	Nivel bajo	Configuración de la LD.
3	Nivel bajo	Reservado.
4	Nivel bajo	A nivel alto, permite puentear el filtro paso bajo en transmisión.
5	Nivel bajo	Reservado.
6	Nivel bajo	Reservado.
7	Nivel bajo	Reservado.
8	Nivel bajo	Reservado.
9	Nivel bajo	Reservado.
10	Nivel bajo	Reservado.
11	Nivel alto	Reservado.
12	Nivel bajo	Reservado.
13	Nivel bajo	Reservado.
14	Nivel alto	Deshabilita el filtro paso alto receptor.
15	Nivel alto	Reservado.





## Capítulo 5

# AFE Driver

### 5.1 Introducción

En este capítulo se procede a realizar una descripción detallada de la arquitectura denominada *AFE Driver*, a la cual se ha referido en capítulos anteriores sin entrar en detalle en la composición del módulo y su funcionamiento. El trabajo de cualquier driver consiste en interactuar con el periférico que se encuentra conectado al sistema, por lo que deben establecer las señales adecuadas en los pines del AFE, ya sea para la configuración del periférico como para gestionar la información que transmite y recibe. En esta descripción se explicarán todos procesos que se establecen en el módulo y el porqué se incorporan. Se definirán todas las arquitecturas internas que componen este *AFE Driver*, y los CORES utilizados junto con sus configuraciones establecidas.

Esta arquitectura se ha diseñado con el fin de poder configurar el AFE como transmisor o receptor de la información a través del canal PLC, dependiendo de la especificación realizada, la cual se establece de manera software mediante la programación del procesador del SoC. La capacidad de poder reconfigurar la arquitectura mediante la programación del procesador provoca un funcionamiento distinto sin necesidad de realizar una implementación nueva del diseño. Como el módulo *AFE Driver* puede trabajar tanto para una configuración transmisora como para una receptora, será necesario hacer un diseño para soportar ambos funcionamientos, y en función de cual se elija, instaurar el adecuado. La problemática de implantar ambas configuraciones en el mismo diseño radica en los diferentes cambios de dominio de reloj a realizar, en la multiplexación del bus bidireccional del AFE, la determinación de qué parte del diseño debe funcionar y cuál debe estar preparada para un posible cambio de configuración, la gestión de las comunicaciones serie con el bus SPI, y la interacción con el procesador a través del IPIF y la interfaz AXI-Lite.

El módulo *AFE Driver* está compuesto por 3 bloques claramente diferenciados: la parte

encargada de configurar el AFE y transmitir o recibir información del mismo; el IPIF para interactuar con la parte software PS y el SPI para toda la configuración que se realiza vía serie al AFE. En la figura 5.1 se puede ver la distribución del módulo *AFE Driver* con sus distintos bloques. A continuación se hace una descripción de los mismos:

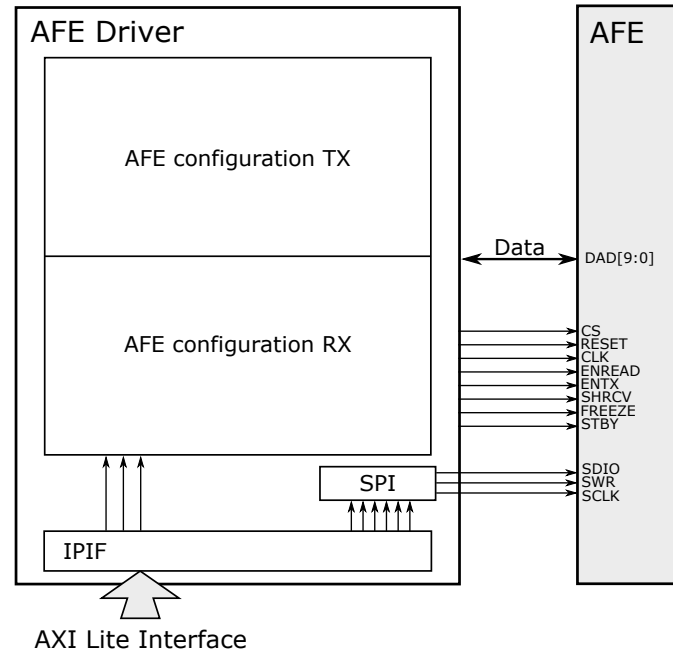


Figura 5.1: Esquema de las partes que componen el módulo *AFE Driver*.

- Bloque I - *Configuración del AFE*. Será la parte encargada de configurar el AFE, realizará una comprobación de los parámetros establecidos en el software para determinar qué señales debe introducir en el AFE para que funcione de la forma seleccionada, y configurará el bus bidireccional como corresponda para transmitir o recibir datos. Realizará un cambio de dominio de reloj debido a que recibe la información desde un dominio de reloj distinto al que lo transmite, ya sea del driver al AFE o viceversa. Para realizar dicho cambio de dominio se utilizarán memorias FIFO. Se configurarán en este bloque las señales CS, RST, CLK, ENREAD, ENTX, SHRCV, DAD[9:0], FREEZE, STBY con unos valores determinados, dependiendo de si se ha establecido una configuración transmisora o receptora. En el apartado 5.2 se detalla la función de cada una de ellas.
- Bloque II - *IPIF*. Este módulo será la interfaz entre la parte software PS y el módulo *AFE Driver* de la parte lógica PL. Será el encargado de traspasarle los valores establecidos en la programación software a la arquitectura de configuración del AFE y al SPI que recibe a través de la interfaz AXI-Lite.
- Bloque III - *SPI*. El módulo SPI se encarga de configurar las características del AFE, transmitiendo la información de los registros que obtiene del IPIF como un flujo de bits hacia el AFE. Para que el AFE interprete de manera adecuada la información, es necesario ajustarse a las especificaciones de su datasheet.

Como ya se ha comentado, el diseño completo que se propone en este trabajo consta de dos *AFE Driver*, uno configurado como transmisor y otro configurado como receptor. En la figura 5.2 se puede ver cómo quedarían los driver en el diseño.

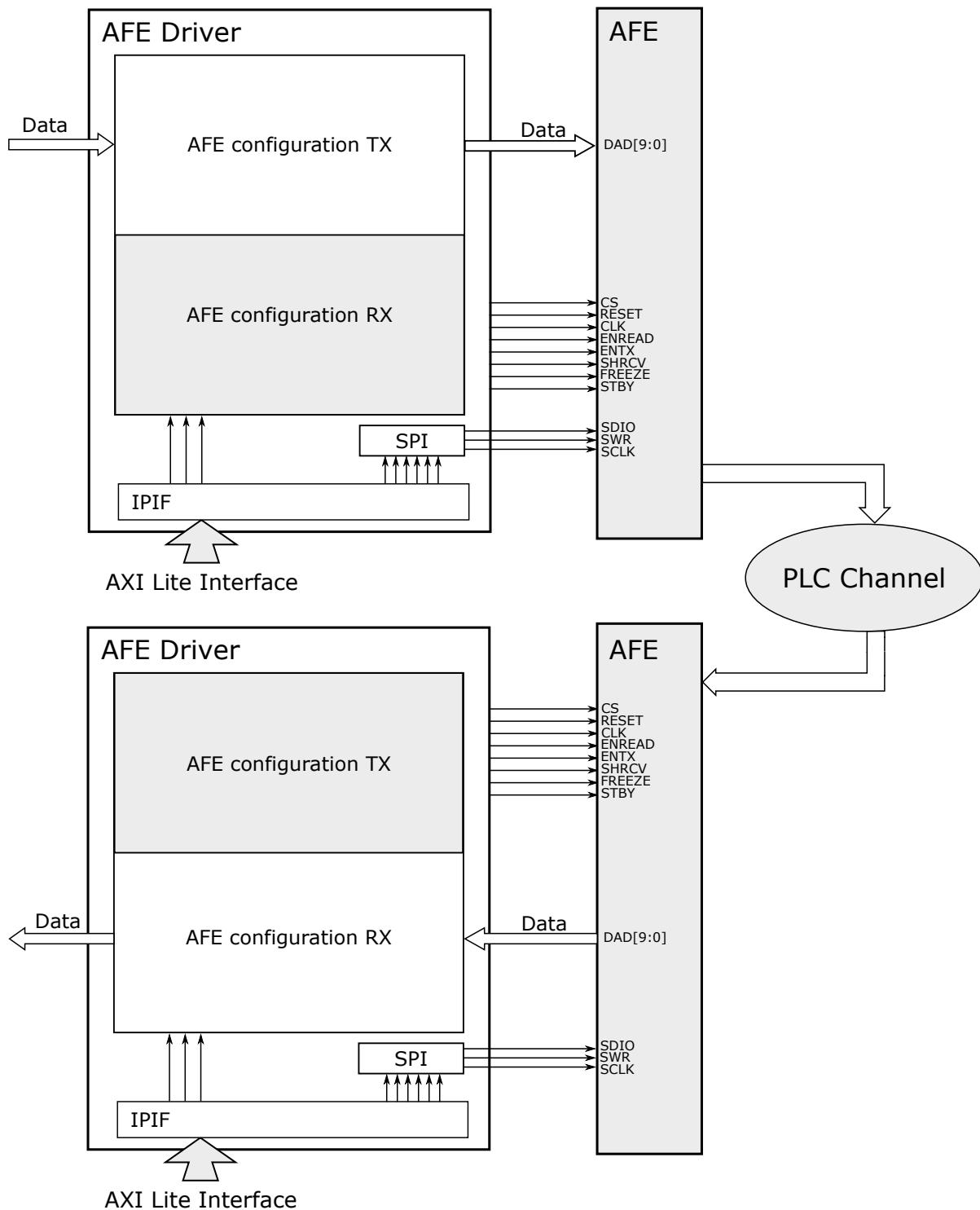


Figura 5.2: Configuración de los *AFE Drivers* del diseño.

## 5.2 Configuración del AFE

### 5.2.1 Descripción de la arquitectura

Este bloque I del módulo *AFE Driver* tiene la finalidad de configurar el AFE estableciendo las señales adecuadas en sus pines para realizar la configuración que se ha prefijado por software y obtener las prestaciones buscadas. Además de establecer las señales de salida para configurar el AFE, también maneja el bus bidireccional de datos del que consta el AFE. Como el sistema funciona a frecuencias superiores de la del AFE, es necesario realizar un cambio de dominio de reloj. Este cambio de dominio de reloj se realizará mediante el establecimiento de memorias FIFO. Como el AFE puede configurarse para transmitir o recibir información, se realiza una arquitectura que sirva para ambas funcionalidades. Por lo tanto, se incluirán dentro del módulo las dos configuraciones de las señales del AFE y los dos cambios de dominio de reloj mediante dos memorias FIFO.

A continuación se describen las señales del AFE que se van a establecer en la arquitectura para las diferentes configuraciones del mismo:

- *CS (AGC Control Signal)*: La señal CS se encarga del control de ganancia adaptativa AGC, este bloque solo existe en el recorrido de recepción del AFE, cuando el AFE se encuentra configurado como receptor. Dicha señal se establece a nivel bajo para establecer una adaptación continua.
- *RST (Reset)*: Es activo a nivel bajo, y debe estar inicialmente activo 100 nanosegundos como mínimo, por lo que se activará inicialmente para una correcta configuración. Esta señal recibe su valor de la parte software PS.
- *CLK (Clock)*: Es la señal de reloj del AFE, según las especificaciones tiene una frecuencia de trabajo de 50 MHz, por lo que se establece un reloj de 50 MHz generado con el MMCM en el pin del AFE correspondiente.
- *ENREAD (Read Enable)*: Esta señal determina la dirección del bus bidireccional de datos DAD[9:0] del que dispone el AFE. Cuando se quiera transmitir por el canal PLC se debe configurar el bus como salida, mandando la información al DAC interno del AFE, para ello, el valor de la señal ENREAD a transmitir al AFE es un nivel alto. Si por el contrario se quiere recibir información del canal PLC a través del ADC del AFE, se establecerá un nivel bajo fijando el bus de datos como entrada.
- *ENTX (Transmit Enable)*: Esta señal habilita la transmisión del AFE cuando se establece a nivel alto. Cuando el AFE funcione en recepción, se fijará nivel bajo.

- *SHRCV (Receiver Power-Down)*: Esta señal sirve para apagar la parte receptora del AFE si se establece un nivel alto en la misma. Cuando el AFE esté transmitiendo se considera buena práctica apagar la recepción. Se recomienda que esta señal valga lo mismo que los valores establecidos en ENREAD y ENTX.
- *DAD[9:0] (Digital-to-Analog and Analog-to-Digital Converter Input/Output)*: Es un bus de datos bidireccional de 10 bits conectado al ADC y al DAC del AFE mediante un multiplexor. Será muy importante establecer este bus como entrada o salida con la señal ENREAD ya descrita, dependiendo de si se quiere transmitir o recibir.
- *FREEZE (AGC Freeze Mode)*: Esta señal sirve para bloquear el AGC de recepción. Por lo que se establecerá a nivel alto en transmisión ya que no se utiliza, y a nivel bajo en recepción.
- *STBY (Standby Control)*: Esta señal sirve para establecer el AFE en standby, se establecerá a nivel bajo para que el AFE esté en continuo funcionamiento.

Si se establece una configuración transmisora en el AFE, se fija a nivel alto la señal ENTX para habilitar la transmisión, a nivel alto la señal ENREAD para establecer el bus bidireccional DAD[9:0] como bus de salida del driver o de entrada en el AFE, a nivel alto SHRCV para apagar la parte de recepción del AFE y a nivel alto FREEZE para apagar el AGC.

Si se establece una configuración receptora en el AFE, se establecerá a nivel bajo la señal ENTX para deshabilitar la transmisión, a nivel bajo la señal ENREAD para establecer el bus bidireccional DAD[9:0] como bus de entrada, a nivel bajo SHRCV para encender la parte de recepción del AFE y a nivel bajo FREEZE para aplicar la ganancia del AGC.

Para transmitir información por el AFE hacia el canal PLC, es necesario establecer los siguientes procesos de configuración a diferentes dominios de reloj, en función de a cuál pertenezca cada acción en el módulo *AFE Driver*.

- AFE Driver Transmisor
  - *Dominio de reloj de 100 MHz*. Si se establece una configuración transmisora en el *AFE Driver*, se conecta la línea de habilitación de escritura de la FIFO de transmisión a la señal que establece el módulo *Interfaz* en función de los flags de llenado de la misma. También desactiva la lectura de la FIFO de recepción para evitar problemas con otras posibles configuraciones.
  - *Dominio de reloj de 50 MHz*. Se transmiten hacia el AFE las señales ENTX, ENREAD, SHRCV, FREEZE. La lectura de la FIFO transmisora se realiza de manera continua debido al consumo de datos a frecuencia igual o menor.

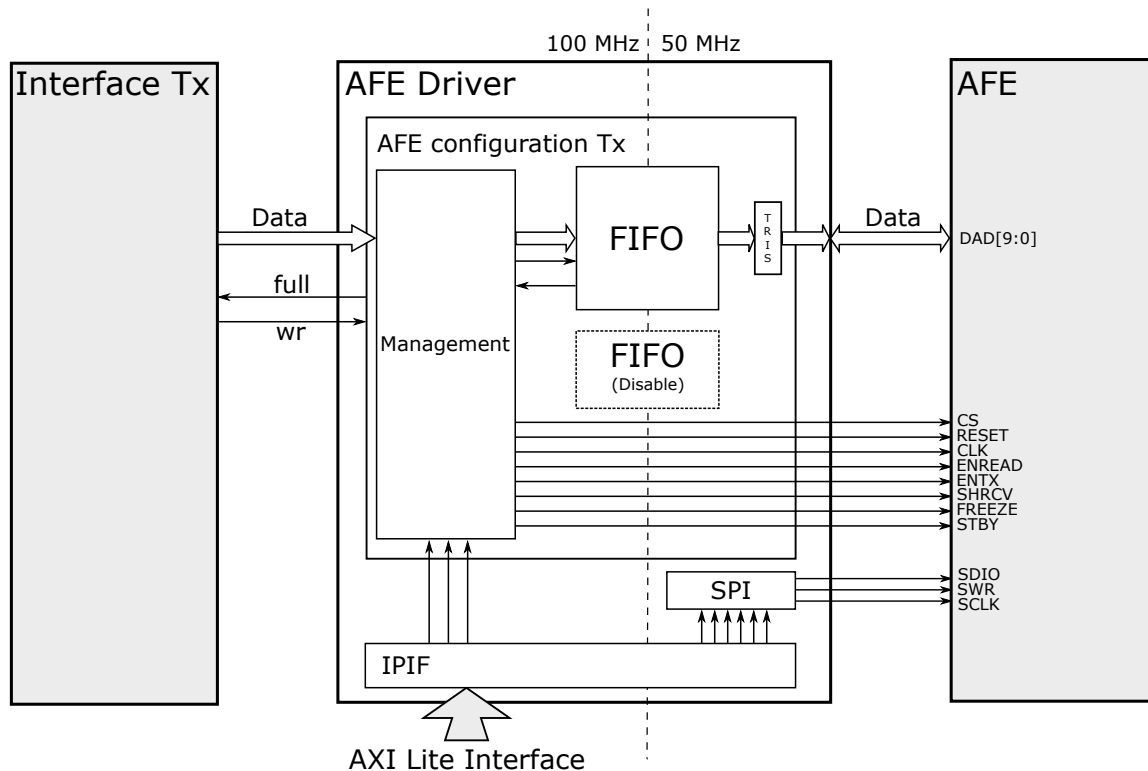


Figura 5.3: Esquema de las partes que componen el módulo *AFE Driver* transmisor.

En la transmisión, el cambio de dominio de reloj es de 100 MHz a 50 MHz. Se requiere una FIFO de 16 posiciones la cual se detallará en el apartado 5.2.3, este es el mínimo tamaño y es suficiente debido a la adaptación que se hace del ancho de banda máximo del AFE. A la salida de la memoria FIFO de transmisión, se introduce un buffer tri-estado. Su funcionamiento consiste en dejar pasar el dato que recibe o establecer la salida en alta impedancia en función de una señal de control. Este buffer es necesario debido a que el bus de escritura y de lectura es el mismo, el bus bidireccional DAD[9:0]; entonces cuando la arquitectura esté en configuración receptora, la transmisora debe establecer alta impedancia para no acceder al bus y que no se produzca una colisión. En la arquitectura receptora no es necesario ya que no realiza escritura en el bus, solo coge los datos de él.

Para recibir información por el AFE desde el canal PLC, es necesario establecer los siguientes procesos de configuración a diferentes dominios de reloj, en función de a cuál pertenezca cada acción en el módulo *AFE Driver*.

- AFE Driver Receptor

- *Dominio de reloj de 100 MHz.* Si se establece la configuración receptora, se conecta la línea de lectura de la FIFO de recepción a la señal que establece el módulo *Interfaz* en función de los flags de vaciado de la FIFO. También desactiva la FIFO de transmisión para evitar problemas con otras posibles configuraciones del *AFE Driver*.

- *Dominio de reloj de 50 MHz.* Se transmiten hacia el AFE las señales ENTX, ENREAD, SHRCV, FREEZE. Estas señales se transmiten con valores de configuración receptora.

En recepción, el cambio de dominio de reloj es de 50 MHz a 100 MHz. Se requiere una FIFO de 16 posiciones. Esto se debe a que la lógica con la cual se obtienen las muestras de la FIFO, es mediante la comprobación del flag *empty* de la FIFO. Como se lee a una frecuencia superior a la que se escribe, con el tamaño mínimo que se puede establecer es suficiente.

- *Dominio de reloj de 50 MHz desfasada 180°*. Como el AFE manda la información al driver en flanco de bajada y el sistema trabaja en flanco de subida, es necesario un reloj desfasado para equilibrar esta diferencia. También activa la línea de escritura de la FIFO de recepción.

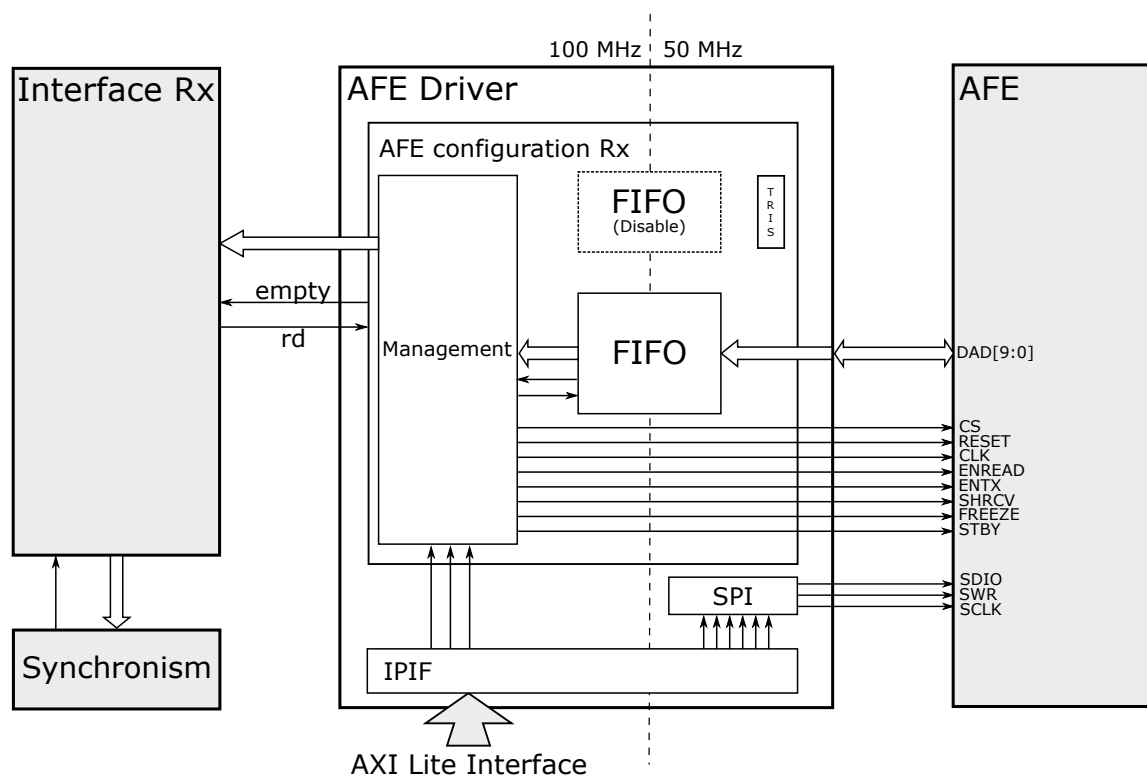


Figura 5.4: Esquema de las partes que componen el módulo *AFE Driver* receptor.

### 5.2.2 Configuración de la arquitectura vía software

Este diseño cuenta con dos módulos *AFE Driver*, uno que transmite la información y otro que la recibe, pero la arquitectura de ambos es la misma. Se consiguen funcionamientos distintos debido a la configuración que se le indica en la programación software. Es por eso que se crean varias señales que provoquen estas diferencias. Estos valores se transmiten mediante la interfaz AXI-Lite al IPIF, que es el bloque que interacciona con los otros

que componen el *AFE Driver* para transmitirle los valores establecidos. A continuación se detallan las variables establecidas para el control de las arquitecturas.

- *in valid*: esta variable establece la configuración del AFE, transmisora o receptora.
- *init conv*: esta variable establece el funcionamiento o la deshabilitación del *AFE Driver*.
- *reset*: El reset del *AFE Driver* se establece mediante software.

En el *AFE Driver* Transmisor se establecen los siguientes valores:

- *in valid*: se establecerá a nivel alto para configurar el *AFE Driver* como transmisor.
- *init conv*: se establecerá a nivel alto para habilitar la arquitectura.
- *reset*: se establecerá un nivel bajo al principio para inicializar la arquitectura y después se fijará a nivel alto cumpliendo con las especificaciones de 100 ns.

En el *AFE Driver* Receptor se establecen los siguientes valores:

- *in valid*: se establecerá a nivel bajo para configurar el *AFE Driver* como receptor.
- *init conv*: se establecerá a nivel alto para habilitar la arquitectura.
- *reset*: se establecerá un nivel bajo al principio para inicializar la arquitectura y después se fijará a nivel alto cumpliendo con las especificaciones de 100 ns.

Cada módulo *AFE Driver* cuenta con un IPIF, y estas variables están por duplicado para configurar de manera independiente ambos drivers. Cada uno tiene una zona de memoria diferenciada en la que escribir. La interfaz AXI-Lite es bidireccional, pero solo se usará en un sentido, no habrá transmisión de información desde la parte lógica PL a la parte software PS. En la función donde se escriben estas variables se establece la dirección de memoria base de ese módulo, el offset correspondiente a cada registro y el valor que se va a establecer. En la figura 5.5 se pueden ver un ejemplo de las direcciones que se establecen a cada módulo para poder escribir y leer.

AXI_AFE_DRV_0	S00_AXI	S00_AXI_reg	0x43C0_0000	64K	0x43C0_FFFF
AXI_AFE_DRV_1	S00_AXI	S00_AXI_reg	0x43C2_0000	64K	0x43C2_FFFF

Figura 5.5: Direcciones de memoria de los módulos *AFE Driver*.



### 5.2.3 Cambio de dominio de reloj

Este diseño cuenta con dos dominios de reloj diferenciados: un dominio de 100 MHz en el que se encuentra todo el diseño referente al tratamiento de datos que se transmiten y reciben de la técnica de acceso al medio FBMC; y otro de 50 MHz con lo referente al AFE físico ya que es su frecuencia de trabajo.

En los diseños en los que se encuentran varias arquitecturas con varios dominios de reloj y se comunican entre ellos, es necesario realizar arquitecturas intermedias como es esta. Una de las funciones de esta arquitectura es realizar dicho cambio de dominio de reloj, con el fin de sincronizar el diseño.

En transmisión al ajustarse el FBMC transmisor a 50 MHz para adaptarse al AFE y no perder información, se escribirá cada dos ciclos de reloj al estar en el dominio de los 100 MHz; la lectura de las muestras almacenadas en dicha FIFO se realiza a una frecuencia de 50 MHz, por lo que el tamaño de la FIFO adecuado debe ser de 16 muestras (el mínimo posible) ya que no se producen a una frecuencia superior. En caso de tener una producción de datos a una frecuencia superior, se encuentra implementada la lógica de control mediante el flag de *full* de la FIFO. En la figura 5.6 se puede ver cómo queda el CORE con todos sus puertos y en la figura 5.7 se pueden ver las señales introducidas en dichos puertos.

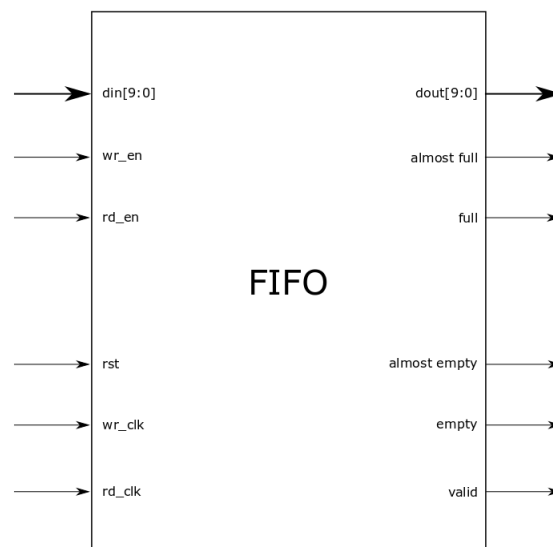


Figura 5.6: Definición de la memoria FIFO que realiza el cambio de dominio de reloj en transmisión.

En recepción, se reciben las muestras a una frecuencia de 50 MHz del AFE, en el dominio de reloj de 50 MHz. La lectura de las muestras almacenadas en dicha FIFO se realiza a una frecuencia de 100 MHz, por lo que el tamaño de la FIFO debe ser el mínimo posible debido a que se consumen a una frecuencia superior de los que producen. En la figura 5.8

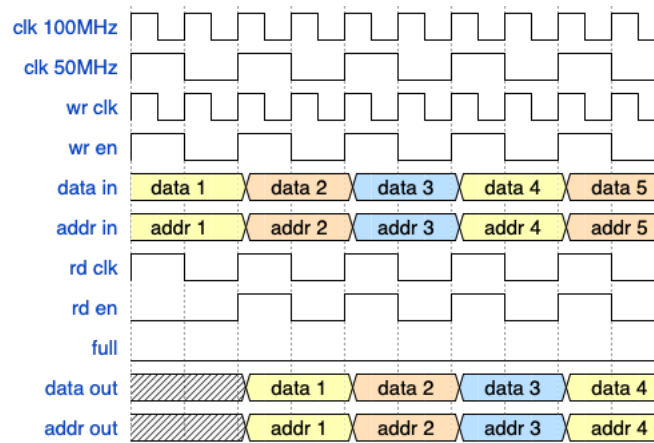


Figura 5.7: Cronograma del funcionamiento de la memoria FIFO en transmisión.

se puede ver cómo queda el CORE de la memoria FIFO con todos los puertos establecidos para su correcto funcionamiento.

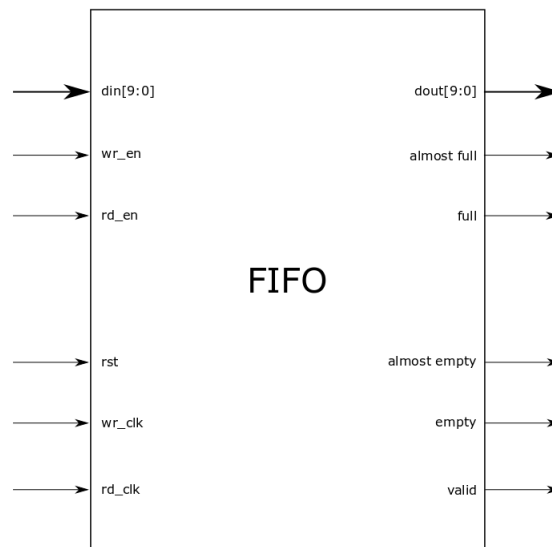


Figura 5.8: Definición de la memoria FIFO que realiza el cambio de dominio de reloj en recepción.

Como la frecuencia de lectura es superior a la de escritura, se necesitará controlar la habilitación de lectura mediante el flag *empty* de la FIFO receptora. En la figura 5.9 se puede ver como en función del flag *empty* se activa la señal de lectura y por tanto se comienza a sacar datos de la memoria FIFO. Como se puede observar, la lectura de cada dato es a 100 MHz, pero al realizarse este control de la lectura la frecuencia media que se obtiene es de 50 MHz, frecuencia con la que se producen los datos por parte del AFE receptor.

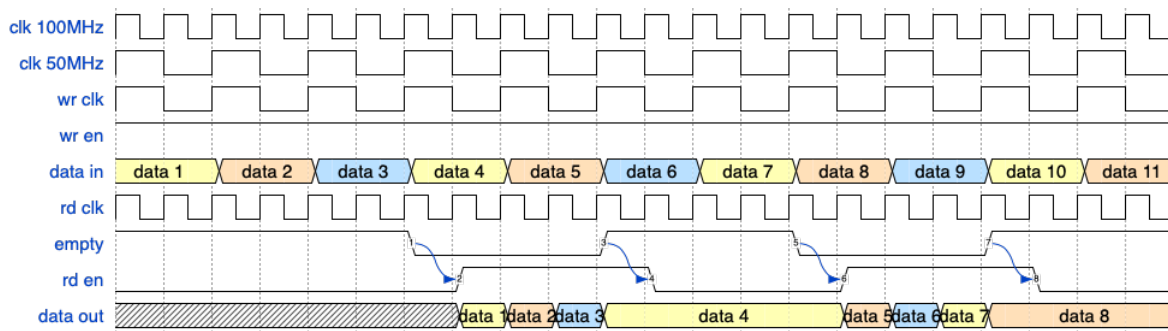


Figura 5.9: Cronograma del funcionamiento de la memoria FIFO en recepción.

## 5.3 SPI (Serial Peripheral Interface)

El bus SPI es un estándar para comunicaciones serie y se va a utilizar para la comunicación entre el *AFE Driver* y el AFE a la hora de escribir los registros del AFE, debido a que se escriben y se leen mediante una única línea. Está compuesto por una línea para la transmisión o recepción del valor de los registros, una línea para indicar si se van a escribir o leer, y una línea para introducir el reloj por el cual se controla el intercambio de información entre ambas partes.

### 5.3.1 Descripción de la arquitectura

Las señales que configura este módulo mediante una transmisión SPI al AFE son las siguientes: SWR, SDIO, SCLK. A continuación se detalla la función de cada una de ellas.

- *SWR (Serial Write-Read)*: Esta señal controla el modo escritura o lectura de la interfaz serie. Si se establece un nivel alto, se escribirán los registros del AFE con los valores transmitidos por el SDIO que se explica a continuación, y con un nivel bajo se leerán los valores establecidos.
- *SDIO (Serial Data Input/Output)*: Es la entrada/salida serie para escribir o leer los registros del AFE.
- *SCLK (Serial Clock)*: Es la línea del reloj serie. Se establece en dicha línea un reloj de 50 MHz.

El funcionamiento de la arquitectura consiste en recibir del módulo IPIF el valor de los registros establecidos en la PS y almacenarlos para transmitirlos bit a bit según corresponda. Hecho esto y habilitada la transmisión, mediante una máquina de estados se establece el mecanismo para enviar los registros. Inicialmente existe un primer estado en el que se realiza una espera para transmitir el registro, estableciendo la línea SDIO por la cual se transmiten los datos como un flujo de bits en alta impedancia. Una vez se termina esta

espera, se procede a transmitir los registros hacia el AFE bit a bit, para ello, se establece la línea SWR a nivel alto y se activa el reloj SCLK para que en su primer flanco de bajada se transmita el primer bit del registro correspondiente. Una vez finalizada la transferencia, se vuelve al estado inicial y se procede a realizar el mismo procedimiento con el siguiente registro, hasta transmitirlos todos. En la figura 5.10 se puede ver una ilustración del funcionamiento de la arquitectura.

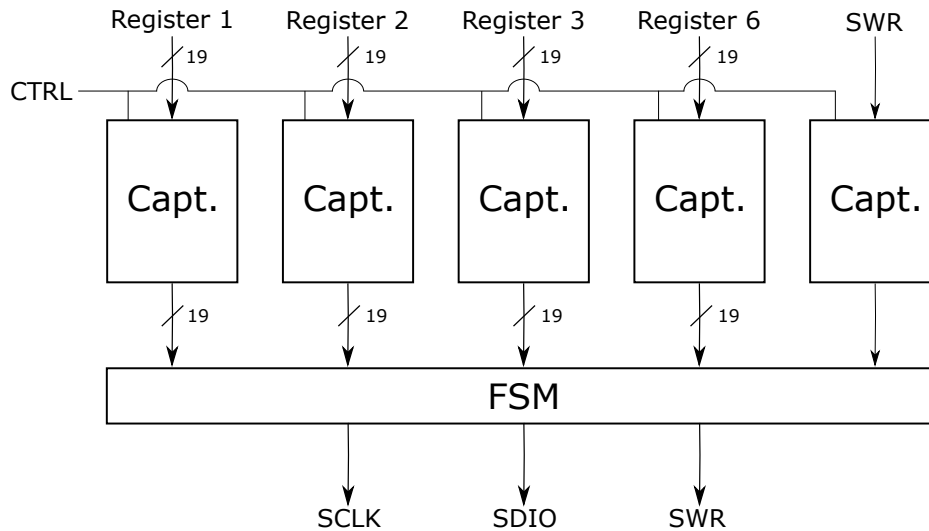


Figura 5.10: Arquitectura del módulo SPI integrado en el *AFE Driver*.

### 5.3.2 Configuración vía software

Este módulo interacciona con el IPIF, que recibe los valores establecidos en la PS, igual que el módulo anterior de configuración del AFE, para transmitir vía serie la información de sus registros que configuran las características del AFE. El IPIF recibe el valor de los registros y la orden de escritura mediante la interfaz AXI-Lite y el módulo SPI los transmite como un flujo de bits hacia el AFE. A continuación se detallan las variables que se transmitirán a la arquitectura:

- *SWR*: Esta variable se asignará a la señal SWR del módulo SPI. Modificarla vía software permite escribir y leer con una simple programación del procesador sin necesidad de implementar de nuevo el diseño o configurar un switch.
- *Registro 1*: Configura el registro 1 del AFE con los valores que se establezcan. En este valor se pondrá concatenado el número de registro, que corresponda con el número 1, y el valor que se quiera establecer como se puede ver en la figura 5.11.

VALOR	0	0	0
-------	---	---	---

Figura 5.11: Configuración del registro 1 del AFE.

- *Registro 2*: Configura el registro 2 del AFE con los valores que se establezcan. En este valor se pondrá concatenado el número de registro, que corresponda con el número 2, y el valor que se quiera establecer como se puede ver en la figura 5.12.

VALOR	1	0	0
-------	---	---	---

Figura 5.12: Configuración del registro 2 del AFE.

- *Registro 3*: Configura el registro 3 del AFE con los valores que se establezcan. En este valor se pondrá concatenado el número de registro, que corresponda con el número 3, y el valor que se quiera establecer como se puede ver en la figura 5.13.

VALOR	0	1	0
-------	---	---	---

Figura 5.13: Configuración del registro 3 del AFE.

- *Registro 6*: Configura el registro 6 del AFE con los valores que se establezcan. En este valor se pondrá concatenado el número de registro, que corresponda con el número 6, y el valor que se quiera establecer como se puede ver en la figura 5.14.

VALOR	1	0	1
-------	---	---	---

Figura 5.14: Configuración del registro 6 del AFE.

- *Control SPI*: Establece si se quiere configurar el AFE con el valor de los registros que se le ha dado en software o si por lo contrario se establecerán los valores por defecto.

Cada bloque SPI está dentro del módulo *AFE Driver* que corresponda e interacciona con su IPIF correspondiente. Cada uno de ellos, como ya se vio en la figura 5.5, dispone de una región de memoria determinada en la que pueden interactuar la parte PL y PS mediante la interfaz AXI-Lite y el IPIF.

### 5.3.3 Resultados experimentales parciales

En este apartado se va a mostrar en primer lugar simulaciones realizadas al funcionamiento del SPI, observando cómo se transmite el valor de los registros bit a bit, tanto si se escribe como si se lee. Se mostrarán simulaciones temporales y señales experimentales, comparándolos con el funcionamiento que se indica en el datasheet para que se haga

correctamente. A continuación, en la figura 5.16 se puede ver una simulación temporal de una transferencia de bits mediante SPI hacia el AFE de 4 registros para poder escribirlos. Como se puede apreciar, la transmisión comienza con la habilitación de la señal control SPI y transmite los registros con el valor que se estableció en el software.

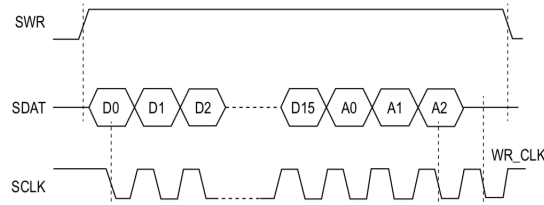


Figura 5.15: Descripción del método de escritura de los registros del AFE dada por el fabricante.

En la figura 5.15 obtenida del datasheet [18] se puede ver cómo el fabricante especifica la forma de escribir los registros del AFE. Si se hace un análisis de la figura, se aprecia cómo se debe establecer un nivel alto en SWR antes de realizar la transferencia de bits hasta que se termina; después se establecerá un valor de alta impedancia para que no interprete valores que no se han establecido. En cuanto al bus serie para transmitir la información, se observa que primero se debe transmitir el número del registro correspondiente empezando desde el bit de menor peso, y a continuación el valor del mismo comenzando también desde el bit de menor peso. En software se establecerá la variable para que cumpla con estos detalles. Por último, el reloj por el que se rige la transferencia debe funcionar durante la transmisión del registro y establecer un flanco de bajada en el que capturar cada bit del registro. Debe apreciarse también que se tiene que establecer un ciclo extra en la señal de escritura SWR y en este reloj serie SCLK.



Figura 5.16: Simulación de la escritura de los registros del AFE.

Se procede a ver en detalle la escritura de los 4 registros para una AFE configurado como transmisor, en la figura 5.17 se aprecia cómo se escribe el registro 1. Se puede apreciar que es el primer registro debido a que en los cuatro últimos flancos de bajada, si no se considera el último flanco extra, hay tres niveles bajos, que se corresponden con el registro 1 como se vio en el apartado anterior en la figura 5.11. En este caso, el valor del registro es un nivel alto en el bit de menor peso y el resto niveles bajos. Esta configuración se realiza cuando se configura el AFE como transmisor, ya que activa el transmisor, desactiva el receptor, activa el DAC y el resto son valores reservados por el fabricante. Si se tuviera que establecer como receptor, se configuraría de manera contraria, como se puede ver en la figura 5.18.

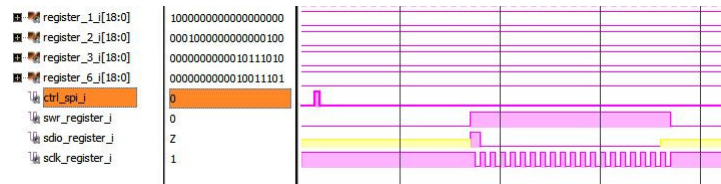


Figura 5.17: Simulación de la escritura del registro 1 del AFE en configuración transmisora.



Figura 5.18: Simulación de la escritura del registro 1 del AFE en configuración receptora.

La figura 5.19 corresponde con la escritura del registro 2. Si se observan los 4 últimos valores descartando el ciclo extra final, se puede ver como se escribe en el bit de menor peso un nivel alto y en los otros dos un nivel bajo, que corresponde con el registro 2, como ya se vio en el apartado anterior en la figura 5.12. El valor del registro es muy limitado, ya que todos los bits salvo uno están reservados por el fabricante. Se puede configurar con el bit de mayor peso el filtro paso bajo de recepción.



Figura 5.19: Simulación de la escritura del registro 2 del AFE.

En la figura 5.20 corresponde con la escritura del registro 3. Si se observa los 4 últimos valores descartando el ciclo extra final, se puede ver como se escribe en el bit central de la parte dedicada al registro un nivel alto y en los otros dos un nivel bajo, que corresponde con el registro 3, como ya se vio en el apartado anterior en la figura 5.13. Se puede configurar con este registro la ganancia si se configura el AFE en recepción.

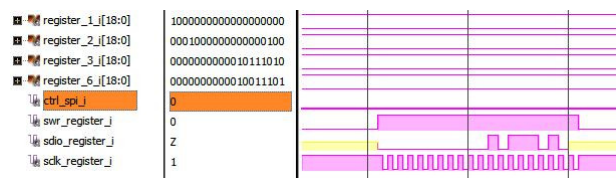


Figura 5.20: Simulación de la escritura del registro 3 del AFE.

En la figura 5.21 corresponde con la escritura del registro 6. Si se observa los 4 últimos valores descartando el ciclo extra final, se puede ver como se establece un nivel alto en el bit de menor peso y mayor peso y el central a nivel bajo, que corresponde con el registro 6, como ya se vio en el apartado anterior en la figura 5.14. Se puede configurar los filtros.



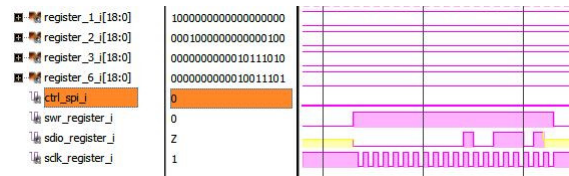


Figura 5.21: Simulación de la escritura del registro 6 del AFE.

En la figura 5.22 se puede ver mediante el CORE Integrated Logic Analyzer (ILA) las capturas experimentales en placa del SPI; como se puede apreciar, son iguales a las simulaciones temporales realizadas, por lo que el resultado es el correcto.

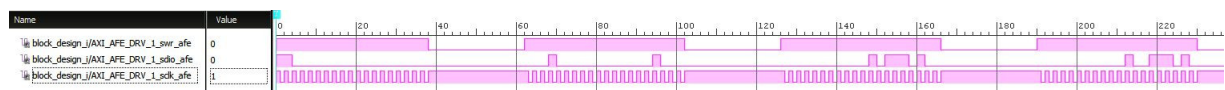


Figura 5.22: Captura experimental en placa de la escritura de los registros del AFE obtenida mediante el ILA.

Mediante el osciloscopio, se van a realizar pruebas para comprobar el correcto funcionamiento del SPI, en la figura 5.23 se puede ver la escritura de los primeros 3 registros. Como se puede observar coincide con lo que se había visto en las simulaciones.

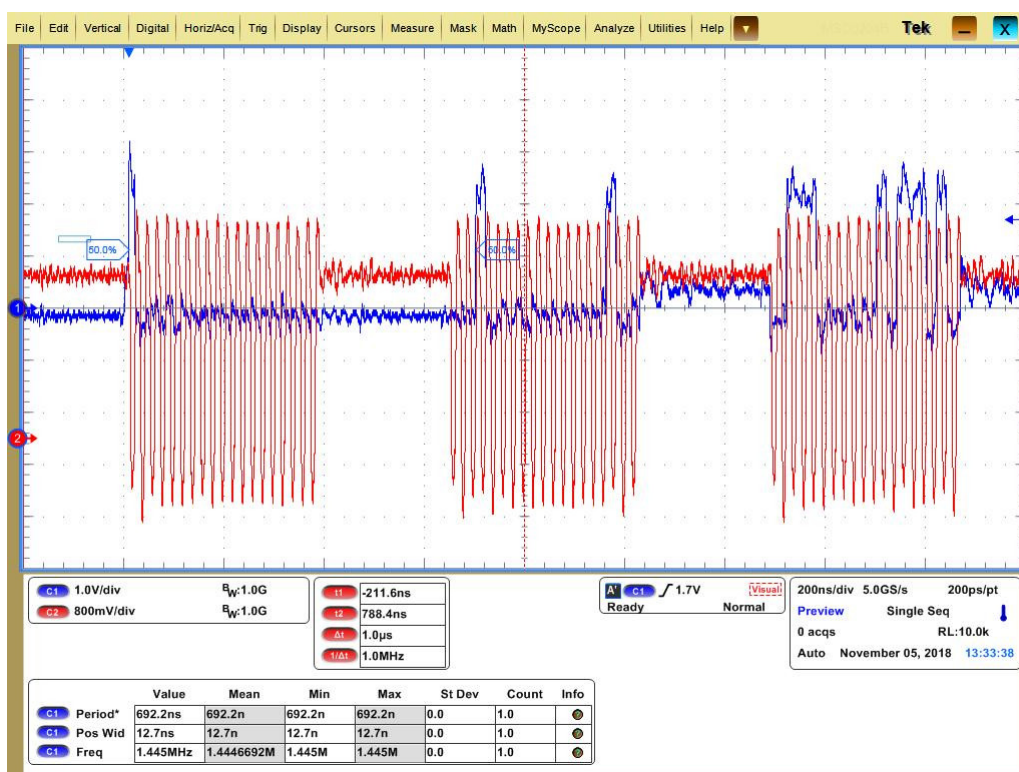


Figura 5.23: Prueba experimental del SPI en el osciloscopio con la escritura de los tres primeros registros.

En la figura 5.24 se pueden ver los dos primeros registros de una manera más detallada. Se puede apreciar cómo se mantiene el nivel correcto en el flanco de bajada de reloj



de acuerdo a las especificaciones. En la figura 5.25 se puede ver la transmisión de los 4 registros en una captura del osciloscopio.

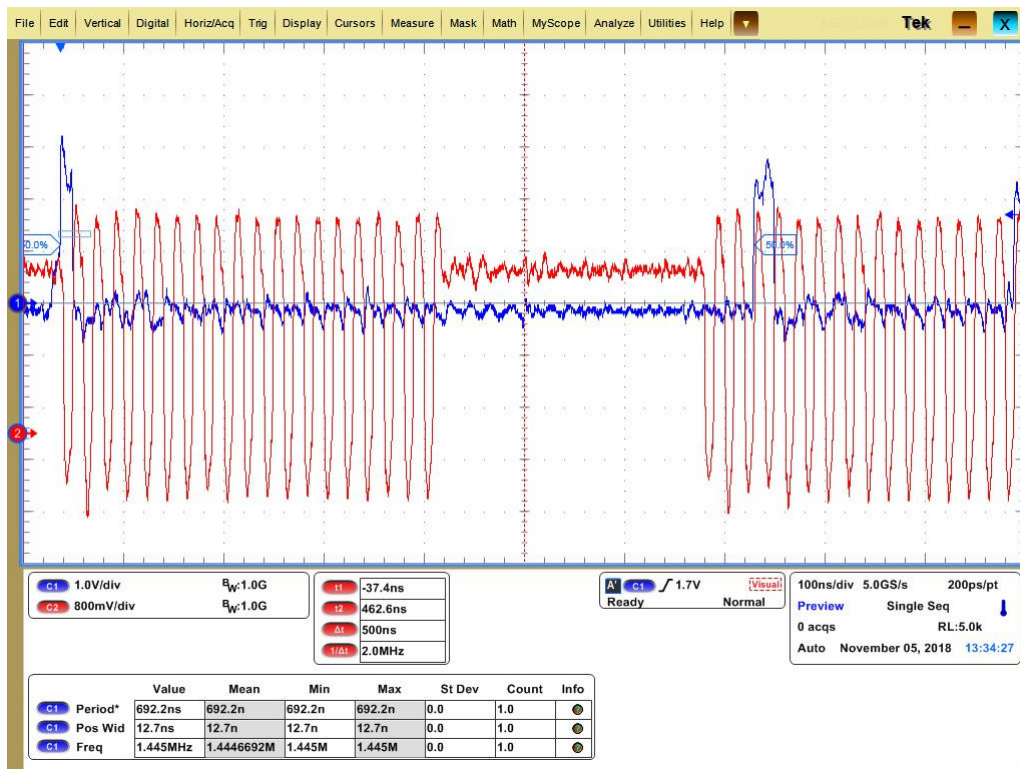


Figura 5.24: Prueba experimental del SPI en el osciloscopio con la escritura de los dos primeros registros.

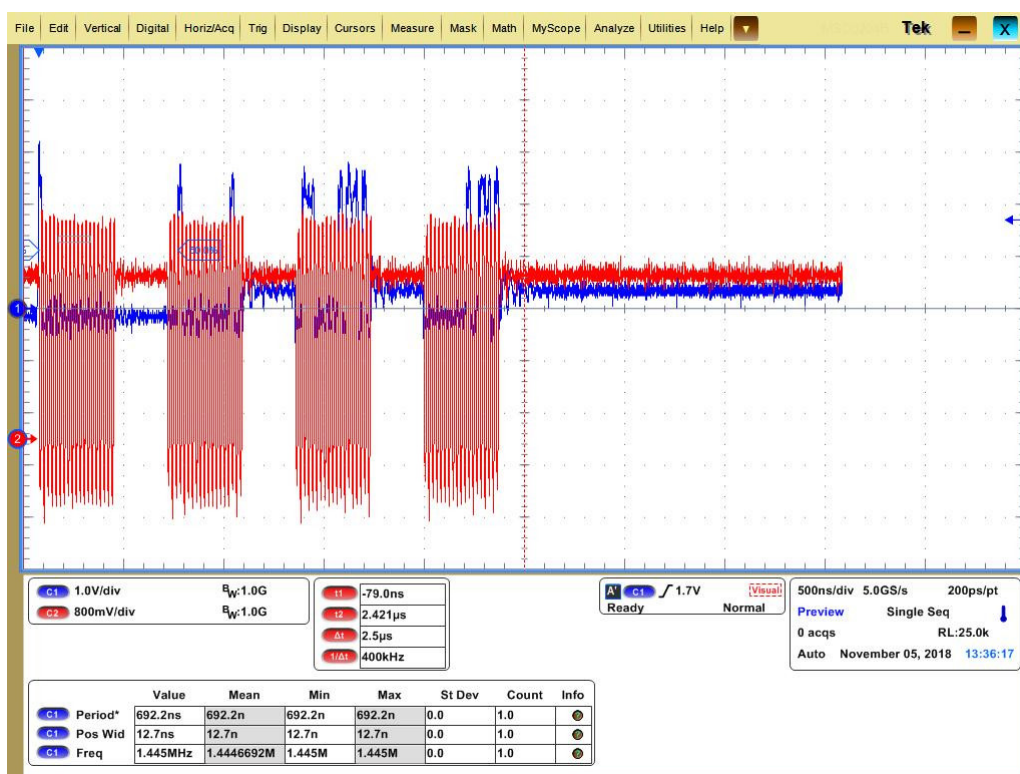


Figura 5.25: Prueba experimental del SPI en el osciloscopio con la escritura de todos los registros.



## Capítulo 6

# Sincronismo

### 6.1 Introducción

El sincronismo se utiliza en los sistemas de comunicación que lo implementan, con la finalidad de conocer en su parte receptora, el inicio de la transmisión de información por parte del transmisor. De esta manera, se puede realizar el tratamiento de los datos recibidos, y conseguir una correcta interpretación de la información que se envió en primera instancia por el transmisor. En este caso, tanto el transmisor como el receptor de la información corresponden con el transmultiplexor FBMC.

El módulo de *Sincronismo* se basa en el estudio teórico desarrollado en [17] y comentado en el capítulo 2. Inicialmente se transmiten dos secuencias piloto, que se almacenan en el módulo *Interfaz* de transmisión que se explicará en el capítulo 7 posterior. Éstas, se transmitirán hacia el AFE realizando un cambio de dominio de reloj en el driver de transmisión para realizar la conversión DA y enviarlas hacia el canal PLC. En recepción, tras realizar el cambio de dominio de reloj inverso en el driver receptor, se almacenan en el módulo *Interfaz* receptor comentado también en el capítulo 7, y se introducirán en el módulo de *Sincronismo*. En este módulo, mediante la operación de correlación que se explicará en el siguiente apartado 6.2, se identifican las tramas de sincronismo enviadas, y por consiguiente se conoce la posición de las tramas de datos que se transmiten justo después.

La correlación cruzada se realiza entre las tramas recibidas desde el canal PLC con las tramas de sincronismo ideales almacenadas en este módulo. Esto provocará que cuando se realice esta operación utilizada para comparaciones, y se introduzca una trama de sincronismo completa en el módulo, al correlar, se obtenga un pico de correlación. A partir de su posición se puede establecer el inicio de los datos.

Este capítulo se desglosará en un primer apartado, en el cual, se expondrá la base teórica del sincronismo que se ha realizado junto con sus operaciones. Después se plantea un modelo en coma fija mediante la herramienta Matlab, para que todas las operaciones se cuantifiquen correctamente sin que se produzcan desbordamientos en la parte entera. Una vez se ha validado, se comentará la arquitectura implementada junto con las simulaciones experimentales del módulo de manera individual.

## 6.2 Estudio teórico

### 6.2.1 Tramas de sincronismo

El método de sincronismo utilizado en el diseño se basa en la transmisión inicial de dos tramas piloto para poder determinar en recepción el inicio de la comunicación. Las tramas utilizadas se obtienen del estudio teórico [17], las cuales, constan de unas características idóneas para la realizar la correlación en la recepción de señales desde canales PLC.

Se utilizarán dos tramas de sincronismo con 512 muestras cada una. La razón por la cual se requieren dos tramas, radica en el hecho de garantizar que al menos una de las dos tramas se introduce completa en el módulo de *Sincronismo* donde se realizan transformadas de 1024 muestras, provocando un pico de correlación.

### 6.2.2 Método de correlación cruzada

La correlación podría definirse como una operación de comparación entre señales, que en este caso corresponde con las tramas de sincronismo enviadas. Se utilizará por tanto dicha operación como una medida del parecido entre las tramas de sincronismo recibidas del canal PLC y las ideales.

En lo referente a la realización de esta operación, el teorema de la correlación establece la posibilidad de obtener la transformada de la correlación entre dos señales a partir de la Transformada de Fourier de las mismas. Para ello, se realiza el producto del conjugado de la transformada de Fourier de una de ellas por la transformada de la otra. En la ecuación (6.1) se puede ver de forma matemática lo mencionado y representado en la figura 6.1, siendo  $x[n]$  las muestras de las tramas ideales e  $y[n]$  las muestras de las tramas recibidas del canal PLC.

$$C_{X,Y}[k] = \mathcal{F}(x[n])^* \cdot \mathcal{F}(y[n]) \quad (6.1)$$

Realizada la correlación, es necesario hacer una transformada inversa para volver al dominio temporal, como se puede ver en la ecuación (6.2). En este dominio, cuando se correla

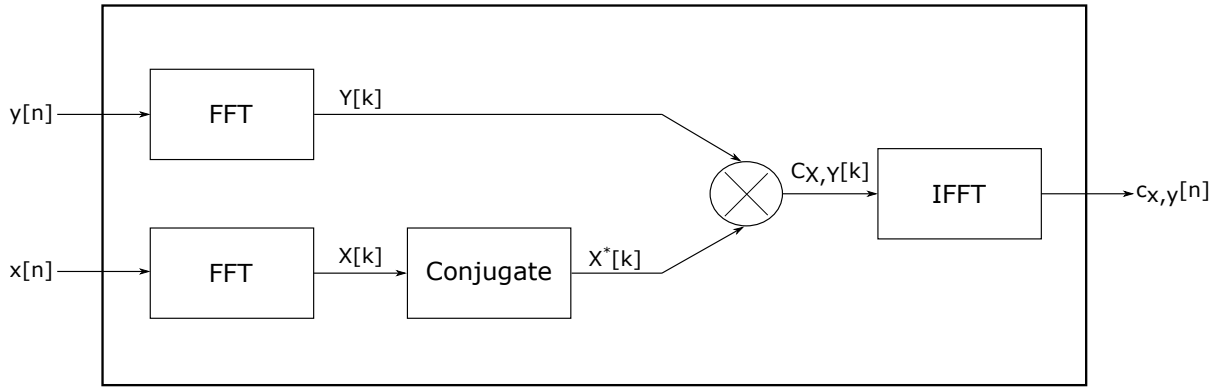


Figura 6.1: Correlación mediante un producto en el dominio de la frecuencia planteada para el sincronismo del diseño.

una trama de sincronismo con ella misma se produce un pico proporcional a su semejanza. Es por tanto necesario detectar el pico de correlación con el cual saber que se han recibido las tramas de sincronismo y fijar el inicio de las tramas de datos.

$$\max(c_{X,Y}[n]) = \max(\mathcal{F}^{-1}(\mathcal{F}(x[n])^* \cdot \mathcal{F}(y[n]))) \quad (6.2)$$

### 6.2.3 Transformada Rápida de Fourier

Como se ha podido ver en el apartado anterior, para realizar la correlación como un producto, es necesario operar en el dominio de la frecuencia, realizando la transformada de Fourier. Se opta por realizar la FFT en detrimento de la DCT. La FFT es una forma de realizar la transformada de una manera más eficiente con respecto a la DCT debido a que utiliza muchas menos operaciones para obtener dicha transformación. Mientras que la DCT requiere  $N^2$  operaciones, la FFT lo reduce a  $N \log_2 N$ , siendo  $N$  el número de puntos de la transformada. Esta disminución supone un menor coste computacional y por tanto una reducción considerable de los recursos. Una vez se ha realizado el producto en el dominio de la frecuencia, es necesario volver al dominio temporal para poder determinar la posición del pico de correlación obtenido. Para ello, se debe hacer la transformada inversa denominada Inverse Fast Fourier Transform (IFFT). Por simplicidad en la realización del diseño y las propiedades de la transformada, se sustituye la IFFT por una FFT, debido a la simetría de la misma. El número de etapas de la FFT/IFFT corresponde con la ecuación (6.3).

$$N_{etapas} = \log_2(N_{puntos}) \quad (6.3)$$

La FFT tiene múltiples versiones con las cuales obtener la transformada. Es por tanto muy importante elegir una adecuada con miras a la implementación de la misma. Para ello, se escoge una arquitectura en donde todas las etapas sean iguales, como se puede ver en la figura 6.2.

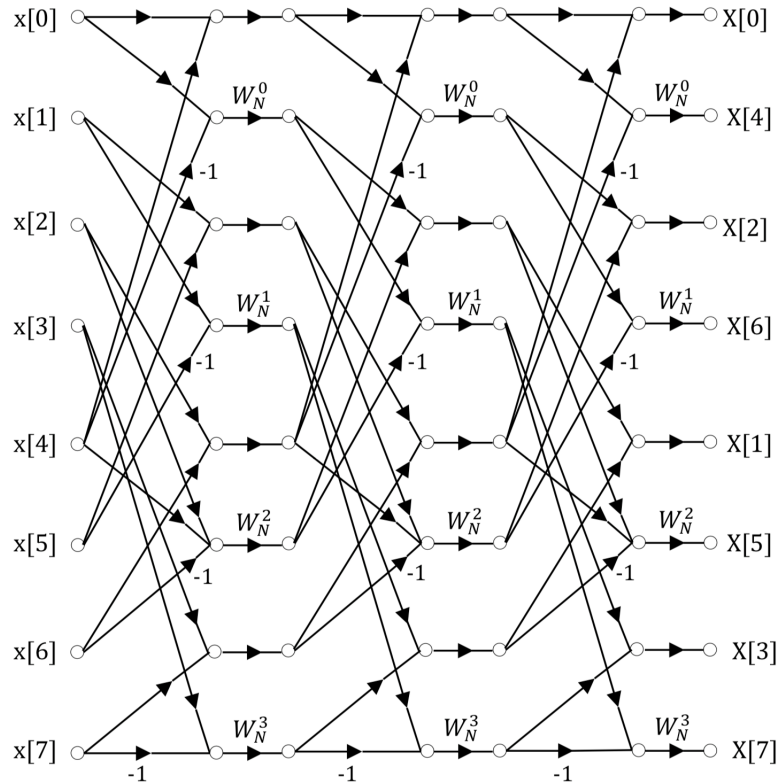


Figura 6.2: Modelo de la FFT utilizada en el diseño cuyas etapas son todas exactamente iguales [19].

Cada etapa cuenta con dos ecuaciones para obtener la entrada hacia la siguiente en función de si el punto es par o impar. En la ecuación (6.4) se puede ver la operación realizada para las impares y en la (6.5) para las pares, donde  $N$  es el número de puntos de la FFT y  $W_n$  es la matriz de los coeficientes de twiddle.

$$x[n] = (x[n] - x[n + N/2]) \cdot W_n[n] \quad (6.4)$$

$$x[n] = x[n] + x[n + N/2] \quad (6.5)$$

Con esta forma establecida, las muestras de la transformada salen desordenadas, pero tiene una solución escalable. Al invertir la posición en representación binaria, se puede ordenar de manera adecuada. En la tabla 6.1 se hace una demostración para una FFT de ocho puntos.

Tabla 6.1: Ejemplo de ordenación de las muestras en el modelo de FFT escogido para 8 puntos.

Proceso de ordenación			
Posición original decimal	Posición original binario	Posición invertida en binario	Posición invertida decimal
0	000	000	0
4	100	001	1
2	010	010	2
6	110	011	3
1	001	100	4
5	101	101	5
3	011	110	6
7	111	111	7

### 6.2.4 Celdas DSP48E1 en las arquitecturas de los dispositivos FPGA

Como ya se ha explicado, cada etapa de la FFT está compuesta por una operación de suma o resta inicial (en función de si la operación es par o impar), y la multiplicación de las muestras que se restan por los coeficientes  $W_n$ . La suma o la resta puede provocar desbordamientos, por lo que es necesario establecer un bit más de parte entera tras realizar las operaciones. Para realizar las multiplicaciones de las que consta la FFT, es obligatorio la utilización de las celdas DSP48E1. Los operandos del multiplicador de la celda DSP48E1 tienen un tamaño de palabra de 25 bits y 18 bits, y se tiene que dejar uno para el posible desborde de las operaciones previas. Por tanto a la entrada, se deben establecer muestras con un tamaño de palabra con un bit menos de lo máximo. En la figura 6.3 se puede ver la composición de la celda DSP48E1, con los tamaños de palabra comentados.

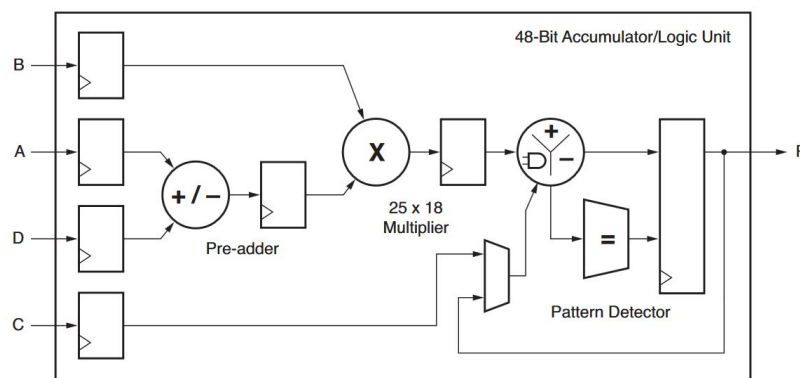


Figura 6.3: Esquema simplificado de la celda DSP48E1 [20].

## 6.3 Modelo en coma fija

El establecimiento inicial de un modelo en coma fija mediante herramientas software como Matlab permite fijar el tamaño de palabra de todas las operaciones que conlleva el diseño, y de esta manera obtener unos resultados adecuados sin que se produzcan desbordamientos y su consiguiente pérdida de información. De manera habitual, cuando se generan estos modelos en coma fija simulando las operaciones de la arquitectura, se hacen acorde al planteamiento teórico de las operaciones que conlleva el diseño, ya que no se sabe el valor de las muestras de entrada. En este caso, al hacer un módulo para unas tramas de sincronismo concretas, con valores dentro de un rango (dependiendo del ruido que añada el canal PLC), se puede realizar un modelo en coma fija que se ajuste correctamente sin producir desbordamiento, de forma experimental. Además, se evita reducir en gran medida los bits de parte fraccionaria como se haría en el modelo ideal teórico, aumentando la precisión y reduciendo el error cometido en la cuantificación de las operaciones. En los siguientes apartados se describen todas las operaciones realizadas en el módulo de sincronismo, la cuantificación teórica ideal, la cuantificación establecida

para la arquitectura y el error que se comete por la utilización de esta representación en coma fija, y los tamaños de palabra establecidos.

### 6.3.1 Cuantificación de las tramas de sincronismo transmitidas

Para proporcionar sincronismo al diseño es necesario la transmisión inicial de unas tramas piloto que hacen posible su correcta identificación en recepción. Por tanto, se generan dos tramas de 512 muestras, que se transmiten previamente a la transmisión de las tramas de datos del módulo FBMC transmisor. Es necesario establecer una cuantificación de tamaño de palabra 10 bits a todas las muestras de la trama, esto se debe al tamaño del bus bidireccional DAD del AFE. Todos los valores de la trama se encuentran entre los valores  $[-2, 2]$ . La asignación de este rango implicaría utilizar un bit de signo más dos bits de parte entera. Con 2 bits de parte entera, conlleva la posibilidad representar el rango  $[-3.9922, 3.9922]$ , muy superior al que es necesario. Por tanto, se opta por establecer solo un bit de parte entera, ya que se representa  $[-1.9961, 1.9961]$ , perdiendo muy poca resolución a cambio de fijar un bit más bit de parte fraccionaria. En la figura 6.4 se pueden ver las dos tramas, tanto para una representación en coma flotante como para coma fija. Se puede apreciar que no existe apenas diferencia con los criterios establecidos.

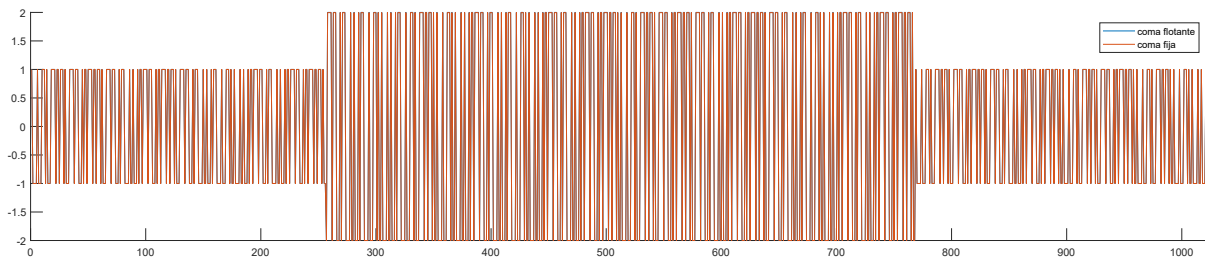


Figura 6.4: Tramas de sincronismo mediante las dos representaciones.

Como se comentará en el capítulo 7, para que los conversores del AFE utilizado realicen una transformación DA/Analog-Digital (AD) adecuada, se requiere realizar el ajuste de rango de conversión, normalmente pasando de complemento a 2 al binario natural empleado por la mayoría de los conversores, tanto de las tramas de sincronismo como de datos. Esto se consigue mediante la negación del bit de mayor peso, provocando que los positivos que tienen como bit de signo un 0, pasen a tener un 1, siendo los mayores del rango establecido; y los negativos con bit de signo un 1, pasen a tener un 0 convirtiéndose en valores menores que los positivos en binario natural. En la tabla 6.2 se puede entender mejor el procedimiento seguido.

Este cambio conlleva pasar a un rango de  $[0, 4]$  como se puede ver en la figura 6.5. Como se sabe que todos los valores pasan a binario natural, no se establece bit de signo, por lo que se dejan 2 bit de parte entera y el resto de parte fraccionaria. Esto hace que se pueda representar el rango  $[0, 3.9961]$ , aunque exista un error de cuantificación por esa reducción de parte entera. Todo esto aparece representado en la tabla 6.3, con el tamaño



Tabla 6.2: Conversión de la información a binario natural para enviarla al conversor DA.

Tramas de sincronismo (10 bits)			
Coma flotante	Coma fija [10, 8]	Binario Natural	Complemento A2
2	1.9961	1111111111	0111111111
1	1	1100000000	0100000000
-1	-1	0100000000	1100000000
-2	-1.9961	0000000000	1000000000

de palabra escogido, su tamaño de parte entera y fraccionaria, el error máximo absoluto y el relativo cometido.

Tabla 6.3: Características de la cuantificación de las tramas de sincronismo en la entrada/salida del AFE empleado.

Representación en coma fija				
Tamaño de palabra	Bits parte entera	Bits parte fraccionaria	Error máximo absoluto	Error Relativo (%)
10	2	8	0.0039	0.1953

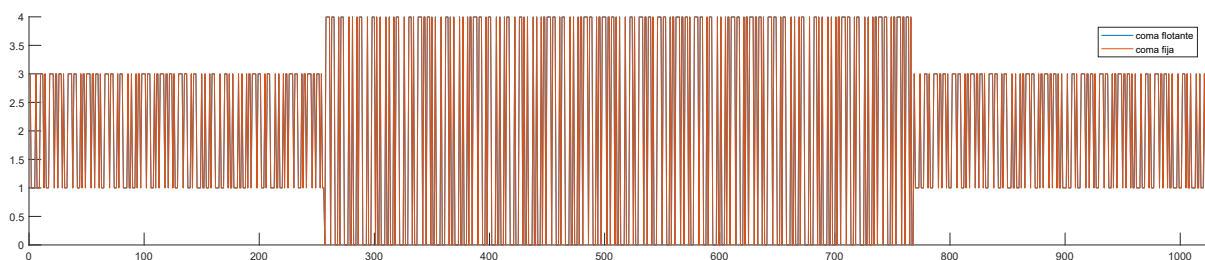


Figura 6.5: Tramas de sincronismo en binario natural a la entrada del AFE empleado.

Realizado todo este proceso, las tramas de sincronismo son transmitidas mediante el canal PLC hacia el AFE receptor, donde se realiza la conversión AD para su correcta identificación. A continuación, se explica la cuantificación realizada en la parte receptora del diseño para el apartado de sincronismo.

### 6.3.2 Cuantificación de las tramas recibidas en el bloque receptor y su FFT

En este modelo se supone un canal ideal, donde se reciben las tramas ideales transmitidas en el bloque receptor. Las tramas se reciben del canal PLC a través del AFE configurado como receptor, mediante su conversor AD, donde las muestrea y las transmite al driver. Cada muestra tiene la misma cuantificación con la que se transmitió, dos bits de parte entera y ocho de parte fraccionaria. Para volver a su representación original, es necesario convertir las tramas de sincronismo de binario natural a complemento a 2. Para ello, se realiza una nueva negación del bit de mayor peso y el resto se mantiene, recuperando los términos negativos.

Para realizar la correlación, como se ha comentado en el apartado 6.2 dedicado al estudio teórico, es necesario realizar la FFT de las tramas de sincronismo recibidas. Como la FFT realizada es de 1024 puntos (el doble de tamaño de una trama de sincronismo, para que al

menos una se introduzca completa), estará compuesta por 10 etapas a razón de la ecuación (6.6). Haciendo un estudio teórico de las operaciones implicadas en la realización de la FFT, se determina que en cada etapa se requiere el aumento de un bit de parte entera debido a que se realiza una suma o resta en cada una de las mismas. De manera teórica, implicaría tener por tanto 2 bit de parte entera más 10 de la realización de la FFT. Como se vio en la explicación del DSP utilizado, se requiere hacer la FFT con una entrada de tamaño de palabra 24 bits. A la salida por tanto, se deberían tener 24 bits de tamaño de palabra, de los cuales, 12 serían de parte entera.

$$N_{etapas} = \log_2(1024) = 10 \quad (6.6)$$

Como se ha podido ver en la figura 6.2, en cada etapa de la FFT se multiplica por los coeficientes de la matriz de transformación lineal  $W_n$ , en los que en su mayoría, cuentan con parte real e imaginaria, lo que implica que los resultados parciales y finales de cada etapa también tengan parte real e imaginaria.

Al ser un módulo ideado para unas tramas concretas (las de sincronismo), no es necesario seguir el modelo teórico que se plantea cuando la entrada no es conocida. Por tanto, se obtienen los máximos de parte real e imaginaria, se realiza el logaritmo en base dos de los mismos y se redondea al entero superior. Además, se añade un bit más debido a que existen valores positivos y negativos. En la tabla 6.4 se puede ver el proceso seguido.

Tabla 6.4: Determinación de los bits de parte entera a la salida de la FFT de las tramas recibidas.

Determinación del tamaño de palabra			
Bits necesarios parte real	Bits necesarios parte imaginaria	Redondeo superior	Bits parte entera
5.3952	0	6	7

Como resultado, en la tabla 6.5 se puede ver la distribución de los bits de la palabra y los errores cometidos por la cuantificación establecida.

Tabla 6.5: Características de la cuantificación a la salida de la FFT de las tramas de sincronismo recibidas.

Representación en coma fija				
Tamaño de palabra	Bits parte entera	Bits parte fraccionaria	Error máximo absoluto	Error Relativo (%)
24	7	17	0.0023	0.0070

A continuación, se representa la parte real en la figura 6.6, y la parte imaginaria en la figura 6.7 en ambas representaciones, donde no se aprecian diferencias significativas. La FFT que realiza Matlab no sigue la misma distribución que la que se realiza en esta arquitectura, por lo que habrá un error entre ambas.

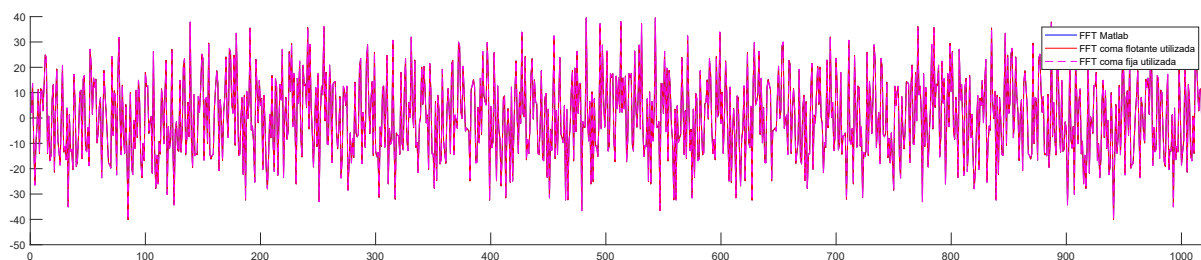


Figura 6.6: Parte real de la FFT de las tramas recibidas por el canal PLC.

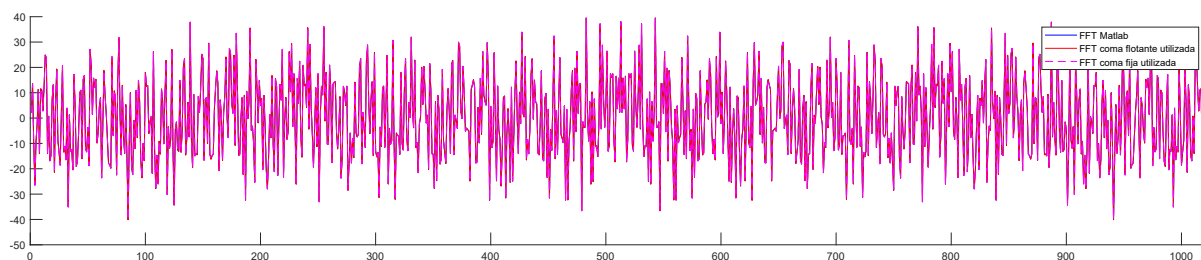


Figura 6.7: Parte imaginaria de la FFT de las tramas recibidas por el canal PLC.

### 6.3.3 Cuantificación del conjugado de la FFT de las tramas de sincronismo ideales

Para realizar la correlación mediante un producto, es necesario realizar el conjugado de la FFT de las tramas de sincronismo. En este caso, son las tramas de sincronismo ideales sin pasar por el canal. Mediante estas tramas se podrá ver la similitud con las transmitidas por el canal y obtener un pico de correlación cuando coincidan. Debido al tamaño de palabra de los operandos del multiplicador de la celda DSP48E1, si se establece un tamaño de palabra de 24 bits para la FFT de las tramas recibidas, para el otro operando, el conjugado de la FFT de las tramas de sincronismo ideales, se debe establecer un tamaño de palabra de 18 bits. De esta manera se evita el uso de una celda DSP48E1 extra en cada operación. Idealmente, al tener unas tramas de sincronismo con 2 bit de parte entera al que se le aplica una FFT de 10 etapas, el resultado debe cuantificarse con un tamaño de palabra de 18 bits con 12 bits de parte entera. Mediante la simulación se ha podido determinar que con 8 bits es suficiente, como se puede ver en la tabla 6.6.

Tabla 6.6: Determinación de los bits de parte entera del conjugado de la FFT de las tramas ideales.

Determinación del tamaño de palabra			
Bits necesarios parte real	Bits necesario parte imaginaria	Redondeo Superior	Bits parte entera
6.1495	6.1290	7	8

El reparto de bits del tamaño de palabra entre la parte entera y la parte fraccionaria, junto con el error máximo absoluto y el relativo que se comete, se representa en la tabla 6.7.

Como resultado, se obtienen la parte real en la figura 6.8, y la parte imaginaria en la figura 6.9 en ambas representaciones, donde no se aprecian diferencias significativas.

Tabla 6.7: Características de la cuantificación del conjugado de la FFT de las tramas de sincronismo ideales

Representación en coma fija				
Tamaño de palabra	Bits parte entera	Bits parte fraccionaria	Error máximo absoluto	Error Relativo (%)
18	8	10	0.0014	0.0068

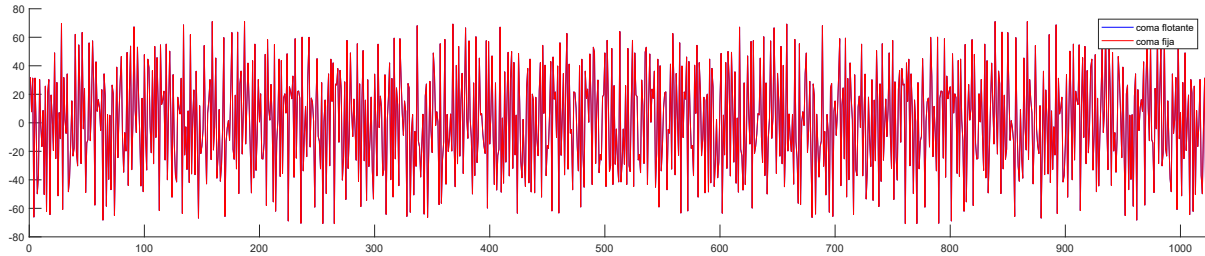


Figura 6.8: Parte real del conjugado de la FFT de las tramas de sincronismo ideales.

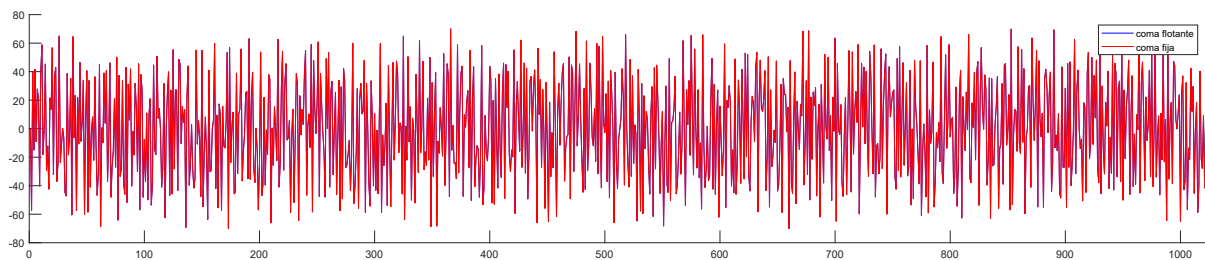


Figura 6.9: Parte imaginaria del conjugado de la FFT de las tramas de sincronismo ideales.

### 6.3.4 Cuantificación de la correlación

Para conseguir la correlación, es necesario la multiplicación de las señales comentadas en los apartados anteriores. La FFT de las tramas recibidas por el canal PLC, como ya se ha comentado tiene un tamaño de palabra de 24 bits con 7 bits de parte entera. El conjugado de la FFT de las tramas de sincronismo tiene un tamaño de palabra de 18 bits con 8 bits de parte entera. Teóricamente, para que no se produzcan desbordes, la multiplicación corresponde con un tamaño de palabra igual a la suma del tamaño de palabra de los operandos, y de parte entera igual a la suma de la parte entera de los operandos. Por tanto se tendría un tamaño de palabra de 42 bits donde 20 son de parte entera. Gracias a las pruebas realizadas, se determina que con 13 bits de parte entera es suficiente. Esto conllevaría a tener un tamaño de palabra de 42 bits con 13 de parte entera. En la tabla 6.8 se puede apreciar la determinación de la parte entera.

Tabla 6.8: Determinación de los bits de parte entera de la correlación.

Determinación del tamaño de palabra			
Bits necesarios parte real	Bits necesario parte imaginaria	Redondeo Superior	Bits parte entera
11.2657	11.4198	12	13

En la figura 6.10 se representa la parte real y en la figura 6.11 la parte imaginaria en ambas representaciones. Como se puede apreciar no existen diferencias significativas.

Al ser necesario realizar una IFFT para volver al dominio del tiempo y obtener el pico de correlación, se cuantifica a 24 bits. En las figuras 6.12 y 6.13 se puede ver que esta

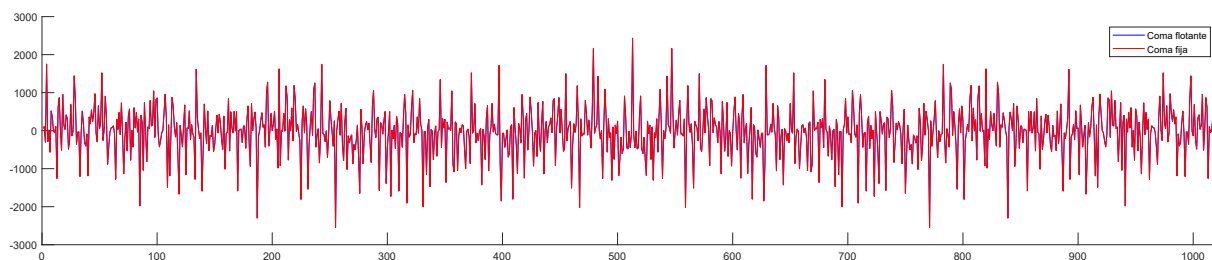


Figura 6.10: Parte real de la correlación cruzada en el dominio de la frecuencia para las distintas representaciones.

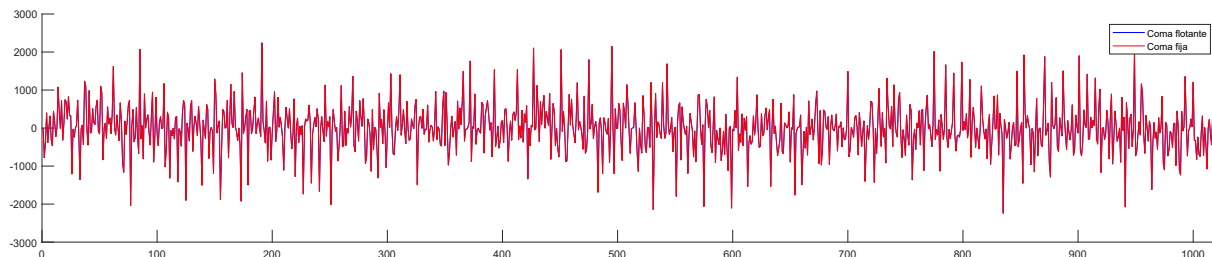


Figura 6.11: Parte imaginaria de la correlación cruzada en el dominio de la frecuencia para las distintas representaciones.

reducción de bits no tiene relevancia. Por tanto, se tendrán 24 bits de tamaño de palabra con 13 bits de parte entera. Todo lo comentado junto con el error cometido se puede ver en la tabla 6.9.

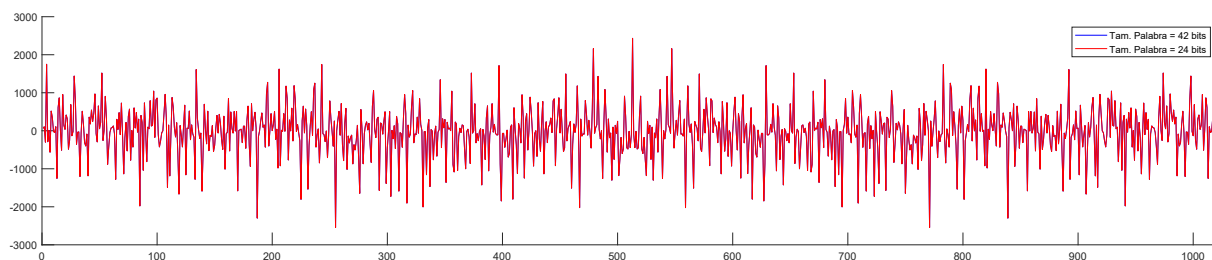


Figura 6.12: Parte real de la correlación cruzada en el dominio de la frecuencia para diferentes cuantificaciones.

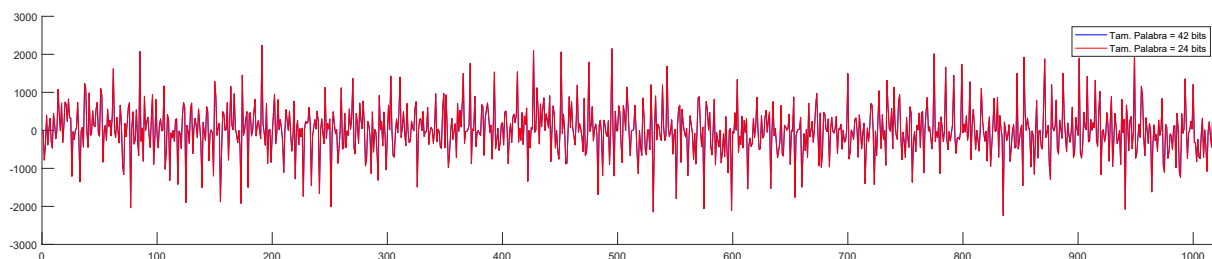


Figura 6.13: Parte imaginaria de la correlación cruzada en el dominio de la frecuencia para diferentes cuantificaciones.

Tabla 6.9: Características de la cuantificación a la salida de la correlación en el dominio de la frecuencia.

Representación en coma fija				
Tamaño de palabra	Bits parte entera	Bits parte fraccionaria	Error máximo absoluto	Error Relativo (%)
24	13	11	0.1640	0.0073

### 6.3.5 Cuantificación de la IFFT de la correlación

Se debería realizar la IFFT para volver al dominio del tiempo, pero como la FFT y la IFFT son simétricas, para simplificar el diseño hardware se reutiliza la FFT que ya se encuentra implementada.

El tamaño de palabra de la multiplicación con la cual se realiza la correlación debe ajustarse al tamaño de palabra de la FFT, que como se ha explicado antes corresponde con 24 bits, por lo que se cogen los de mayor peso, contando con todos los de la parte entera y perdiendo algo de precisión en la parte decimal. Como ya se comentó anteriormente, cada etapa de la FFT aumenta la parte entera en un bit. Por tanto a la salida, si se parte de un tamaño de parte entera de 13 bits, en la cuantificación teórica se tendría un tamaño de palabra de 24 bits con 23 bits de parte entera. Como se ha realizado una simulación, se ajusta a los valores obtenidos como se puede ver en la tabla 6.10, donde se aprecia que con 21 bits de parte entera son suficientes. Con esta cuantificación, el error cometido se representa en la tabla 6.11.

Tabla 6.10: Determinación de los bits de parte entera de la correlación en el dominio del tiempo.

Determinación del tamaño de palabra			
Bits necesarios parte real	Bits necesario parte imaginaria	Redondeo Superior	Bits parte entera
19.2674	2.8580	20	21

Tabla 6.11: Características de la cuantificación de la correlación en el dominio temporal.

Representación en coma fija				
Tamaño de palabra	Bits parte entera	Bits parte fraccionaria	Error máximo absoluto	Error Relativo (%)
24	21	3	54.0604	1.2109

Cuando se introduce la trama de sincronismo sin retardo ninguno, el pico de correlación aparece en la primera posición, como se puede ver en la figura 6.14.

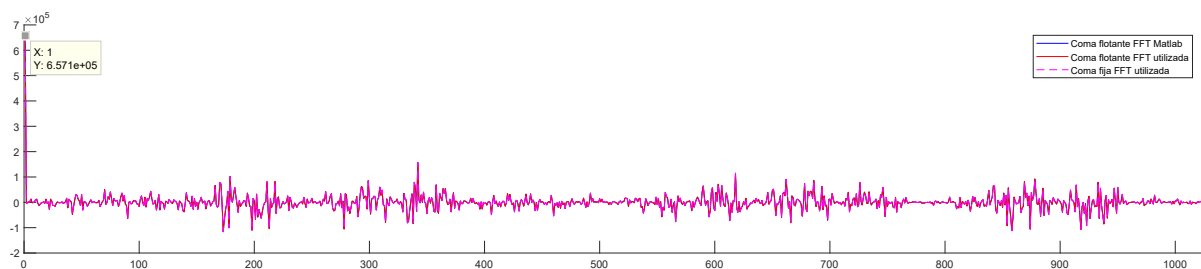


Figura 6.14: Parte real de la correlación en el dominio del tiempo sin retardo en las tramas introducidas.

Si se introduce un retardo de 300 posiciones, el pico aparece en la posición 1024 menos 300, equivaliendo con la 724 (véase la figura 6.15).

A partir de las simulaciones, es necesario establecer una lógica adecuada para determinar la posición del inicio de los datos. Introduciendo las tramas sin retardo en el módulo, es decir, las 1024 posiciones las ocupan las dos tramas de sincronismo de 512 muestras; el

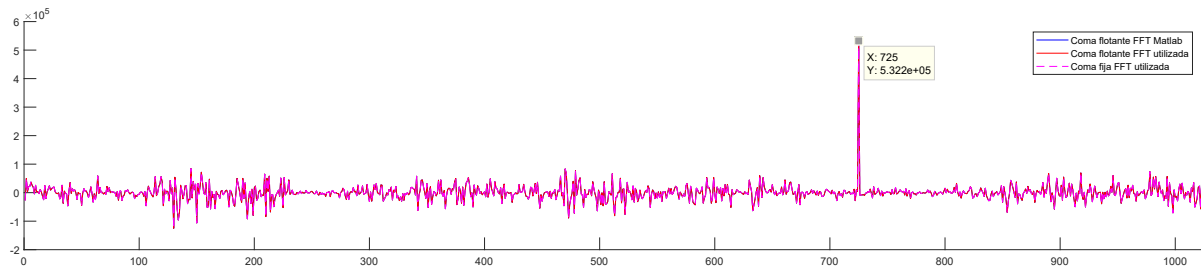


Figura 6.15: Parte real de la correlación en el dominio del tiempo con retardo en las tramas introducidas.

pico aparece en la primera posición, con retardo igual a cero, y los datos 1024 posiciones después (debido a que previamente están las dos tramas de sincronismo de 512 posiciones). Como esta situación no ocurrirá, se plantea el siguiente escenario: si se introduce retardo, es decir, entra una parte de la primera trama de sincronismo y solo entra completa la segunda trama, el pico aparece en la última posición de la segunda trama, marcando el inicio de los datos, como se puede ver en la figura 6.16 para un retardo de 300 posiciones.

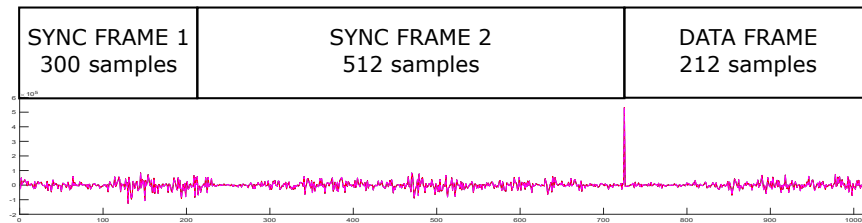


Figura 6.16: Determinación de las 1024 muestras entrantes al módulo de sincronismo en función del pico de correlación.

Con todo lo comentado, se opta por tanto por una solución de compromiso. Debido a que la posibilidad de que en la ventana de la FFT de 1024 posiciones entren las dos tramas de sincronismo de 512 muestras cada una, se opta por determinar el inicio de la transmisión de datos a partir del pico de correlación cruzada.

## 6.4 Definición de la arquitectura del sincronismo

Establecido el modelo en coma fija de las operaciones del diseño, es momento de realizar la implementación de la arquitectura. En la figura 6.17 se representa como un diagrama de bloques, el cual se irá explicando secuencialmente en este apartado. Como se ha comentado en el apartado anterior, es necesario realizar la multiplicación entre la transformada de Fourier de las tramas recibidas del canal PLC y el conjugado de la FFT de las tramas de sincronismo ideales. Estas últimas, al ser valores constantes, se opta por almacenarlas en una memoria directamente para no realizar todas las operaciones necesarias y gastar recursos innecesarios, pero sí será necesario implementar la FFT de las tramas recibidas, ya que varían en función del canal.

Se realiza la arquitectura de la FFT descrita en el estudio teórico, en donde todas las etapas de la misma son iguales. Por tanto, realizada una etapa y replicándola  $\log_2(N)$



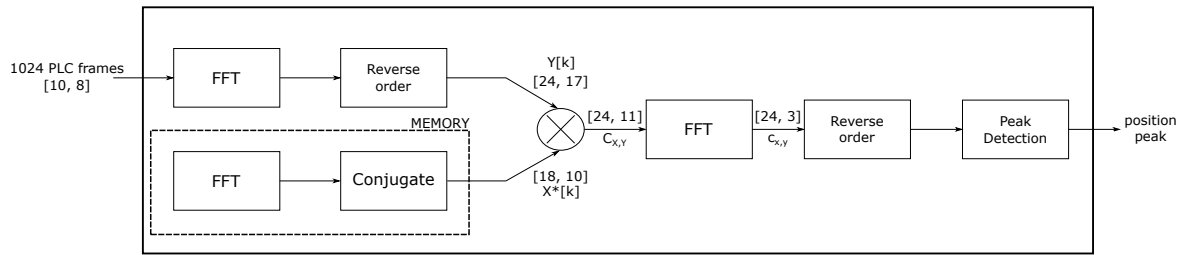


Figura 6.17: Arquitectura del módulo de *Sincronismo*.

veces se obtiene la transformada. La desventaja que presenta la arquitectura radica en las muestras de salida, ya que no están en orden, por lo que es necesario establecerlo antes de operar con el resultado. Su ordenación se implementa mediante la solución comentada en el apartado teórico, que consiste en invertir la posición de salida en representación binaria, obteniendo la posición correcta. Por tanto, a la salida de la FFT se establece la lógica denominada en la figura 6.17 como *Reverse Order*.

La FFT tiene definido un tamaño de palabra para sus operaciones de 24 bits. Al utilizar DSP48E1 que contienen multiplicadores con tamaño de palabra 25x18, los operandos deben de tener un tamaño de palabra de 25 bits y 18 bits. Como la FFT que se realiza a las muestras obtenidas del canal PLC se le establece un tamaño de palabra de 24 bits (más un bit de la operación previa a la multiplicación, 25 bits), para el conjugado de la FFT de las tramas de sincronismo será necesario establecer un tamaño de palabra de 18 bits.

Cuantificadas ambas partes, es necesario realizar una multiplicación. El tamaño de palabra resultante obtenido es la suma del tamaño de palabra de ambas. Se obtendrá un tamaño de 42 bits con 13 bits de parte entera debido al ajuste que se realizó mediante el modelo de Matlab.

Para pasar al dominio temporal se debe realizar una IFFT; debido a las propiedades de la transformada de Fourier, se sustituye por la FFT debido a que son simétricas, evitando añadir más complejidad al diseño implementando una nueva transformada. Inicialmente, se debe cuantificar ese tamaño de palabra (42 bits) al tamaño de palabra de la FFT, que corresponde como ya se mencionó anteriormente a 24 bits. Para ello, es necesario mantener toda la parte entera y perder resolución de la parte fraccionaria. Realizada la FFT, es necesario ordenar las muestras de salida al igual que en la realizada anteriormente.

En el dominio temporal, es necesario implementar una lógica para la detección del pico de correlación en caso de que lo hubiese. Se implementa por tanto un umbral basado en simulaciones experimentales y detección de máximos que superen el umbral. La posición



del pico permite saber la distribución de las 1024 muestras introducidas en el módulo de *Sincronismo*, y por tanto conocer el inicio de los datos.

Este módulo, debe interaccionar con el módulo *Interfaz*, en el cual se han ido almacenando todas las muestras en bloques de 1024 posiciones, correspondiendo con el tamaño del bloque introducido en el módulo de *Sincronismo*. Con la información obtenida del pico de correlación, y la distribución en bloques de todas las muestras recibidas del canal PLC, se determina el inicio de la trama de datos.

## 6.5 Pruebas experimentales

Generada la arquitectura del módulo de *Sincronismo*, se valida el módulo de manera particularizada antes de añadirla al diseño completo. Para ello, se establecen dos memorias de 512 posiciones cada una para guardar las tramas de sincronismo. Se procede a realizar una entrada continuada de las tramas simulando las que se recibirían del canal PLC. En la figura 6.18 se puede ver una de las dos salidas del módulo de sincronismo que corresponde con la correlación en el dominio del tiempo. Se aprecia como el pico de correlación aparece en la primera posición al introducirse completa y sin retardo en el módulo.

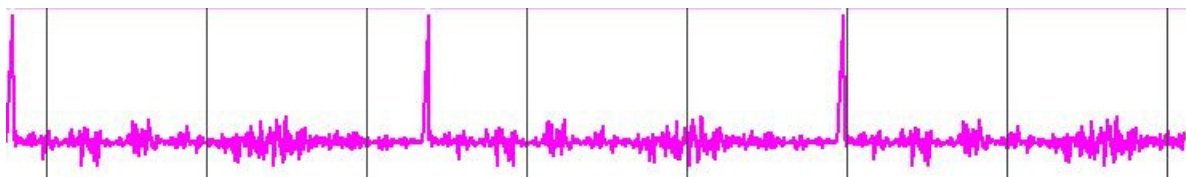


Figura 6.18: Prueba experimental del módulo de *Sincronismo*.

En el caso de hacer un banco de pruebas con parte de la primera trama de sincronismo, la segunda trama de sincronismo completa con sus 512 posiciones y el restante con datos de la trama hasta llegar a las 1024 posiciones, el pico de correlación aparece en la posición 1024 menos el número de muestras de la primera trama, que correspondería con el retardo que sufre la segunda trama hasta entrar completa.



## Capítulo 7

# Módulo de Interfaz

### 7.1 Introducción

Este módulo se introduce en la arquitectura para la gestión y control de las distintas partes que componen el diseño. En primera instancia, su función radica en adaptar el transmisor y receptor FBMC a los drivers que manejan el funcionamiento de los AFE. Esto implica realizar tanto el tratamiento de los datos como la lógica de control. Y en segunda instancia, interactuar con el módulo de sincronismo para determinar el inicio de la comunicación entre el transmisor y receptor del transmultiplexor FBMC. Éste es por tanto, un módulo de gran importancia dentro del diseño.

Aunque en la figura del diseño propuesto en el capítulo 3 se representa el transmultiplexor FBMC como un único módulo, en el diseño real está compuesto por varios bloques de transmisión y de recepción para poder implementarlos por separado en diferentes SoC, aunque en este caso, la transmisión y la recepción se realiza en el mismo. Debido a que se implementa un sistema de comunicación completo, se requiere realizar una adaptación tanto de la transmisión como de la recepción; por ello, el módulo *Interfaz* cuenta con una parte dedicada al acoplamiento de cada sentido de la comunicación. Como este diseño se realiza para una comunicación en la que se transmite desde el SoC al canal PLC y se recibe en el mismo SoC, ambas partes (la interfaz transmisora y receptora) se realizan en un único bloque, el módulo *Interfaz*. Si se busca comunicar dos SoC diferentes, se deberá realizar una distinción entre la transmisión y recepción, pero en este caso se puede establecer todo en un único módulo. En la figura 7.1 se puede ver cómo interactúan las diferentes partes del módulo *Interfaz* con el resto del diseño.

Partiendo desde el inicio de la comunicación, las principales funciones que realiza este módulo en transmisión comienzan con una primera cuantificación del tamaño de palabra, el transmisor FBMC envía tramas compuestas por subportadoras que tienen un tamaño

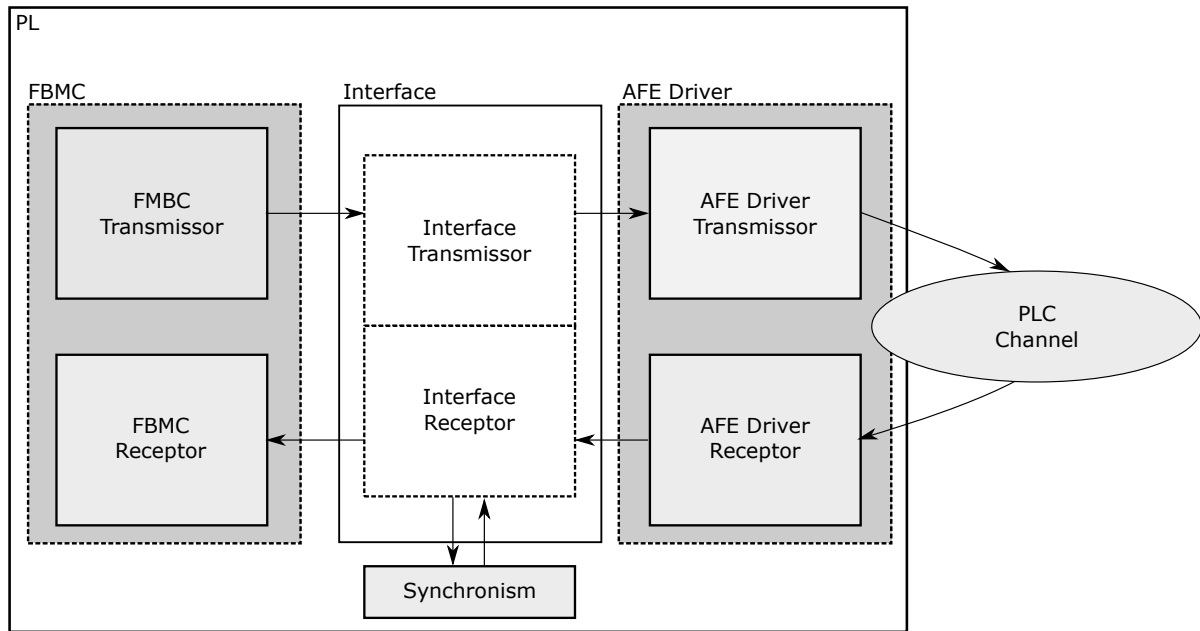


Figura 7.1: Esquema de la interacción del módulo *Interfaz* con el resto del diseño.

de palabra superior al tamaño del bus del AFE, por lo que se deberá ajustar con la consiguiente pérdida de precisión. Una vez cuantificados los tamaños de palabra, se almacenan las subportadoras en una memoria FIFO mientras se transmiten las tramas de sincronismo, que están almacenadas en una memoria BRAM en el módulo *Interfaz*. Se debe realizar una lógica de control para ambas memorias debido a que trabajan en el dominio de reloj de los 100 MHz, y se deben transmitir muestras al *AFE Driver* a una frecuencia de 50 MHz. Esta parte transmisora del módulo *Interfaz* también controla la escritura de la FIFO de transmisión del *AFE Driver*. Mediante el flag *full*, se habilita o deshabilita la escritura, pero, como ya se comentó en el capítulo 5 dedicado al *AFE Driver*, este diseño está adaptado para tener 50 MHz a la entrada del mismo, por lo que no se tendrán problemas de desbordamiento en dicha memoria FIFO. En caso de establecer una frecuencia superior, sí será necesario este control y por eso se establece la lógica necesaria.

En su función dentro del bloque receptor, se encarga de realizar inicialmente una lógica de control para leer las muestras que recibe del driver receptor; esto se debe a que el sistema funciona a 100 MHz y se reciben las muestras del AFE a 50 MHz, por lo que será necesario un control a la hora de leerlas. Estas muestras obtenidas se almacenan en paralelo en dos memorias: una para proporcionar bloques de muestras al módulo de *Sincronismo* y determinar cuál es la primera subportadora, y otra que sirva de buffer para almacenar todas las muestras recibidas y poder obtener la indicada por el sincronismo. Una vez encontrada, se busca dentro del buffer de muestras y se trasmite al receptor FBMC a partir de la misma. Una vez se comienzan a transmitir correctamente las tramas ordenadas desde la memoria, es necesario almacenarlas en dos memorias DUAL PORT. Esto se debe a dos motivos: se reciben las subportadoras a 50 MHz y se deben mandar al receptor

FBMC a 100 MHz, y se deben transmitir en orden inverso del que se transmitieron. A continuación se explicará todo lo comentado de maneras más detallada.

## 7.2 Interfaz Transmisor

El módulo *Interfaz* para transmisión se encarga de adaptar el transmisor FBMC con el driver del AFE encargado de gestionarlo como transmisor. Inicialmente, el bloque FBMC comienza la transmisión de tramas de datos y habilita los flag *ce* y *we*, procediendo a la transmisión de las tramas de sincronismo almacenadas en una memoria BRAM de la interfaz. En la figura 7.2 se puede ver la conexión de estas señales. A su vez, se transmiten las tramas de datos desde el FBMC transmisor hacia una memoria FIFO de la interfaz, pero antes, es necesario comenzar realizando una cuantificación del tamaño de palabra de cada subportadora, denominado en la figura 7.2 como *quantizer*. Se recibe cada subportadora con un tamaño de palabra de 24 bits, por lo que se requiere reducirlo al tamaño del bus bidireccional DAD del AFE (comentado en los capítulos 4 y 5), de tamaño de palabra 10 bits, provocando una pérdida de precisión en la información de cada subportadora. Se tomarán los 10 bits de mayor peso y el resto se descartarán. Además, como ya se ha comentado en el capítulo 6 para las tramas de sincronismo, es necesario transformar las tramas recibidas del transmisor FBMC a binario natural, para adaptar los datos a las etapas de conversión DA/AD del AFE. Por tanto, se negará el bit de mayor peso y el resto de la palabra se mantendrá.

Hecha la cuantificación y pasadas las tramas de datos a binario natural, se almacenan las subportadoras en la memoria FIFO de la interfaz como se ha comentado previamente. Estas tramas son almacenadas hasta que se transmiten las tramas de sincronismo. Cuando se termina, se conmuta para mandar las subportadoras almacenadas en la FIFO y las próximas que se transmitan desde el módulo FBMC. En la figura 7.2 se puede apreciar cómo una vez se transmiten las tramas de sincronismo desde la BRAM, se conmuta a la memoria FIFO para transmitir las subportadoras almacenadas.

Profundizando en estos primeros pasos, la memoria BRAM almacena dos tramas de sincronismo, que cuentan con 512 muestras cada una, por lo que el tamaño de la memoria BRAM será de 1024 posiciones. Al colocarse una memoria FIFO en paralelo para almacenar las subportadoras de las tramas de información mientras se transmiten las tramas de sincronismo al *AFE Driver*, el tamaño de dicha FIFO debe ser como mínimo el número de muestras de las tramas de sincronismo, que son 1024.

El transmultiplexor FBMC, como ya se comentó en el capítulo 2, está implementado para cumplir el estándar IEEE 1901-2010 de comunicaciones PLC. Esto significa que

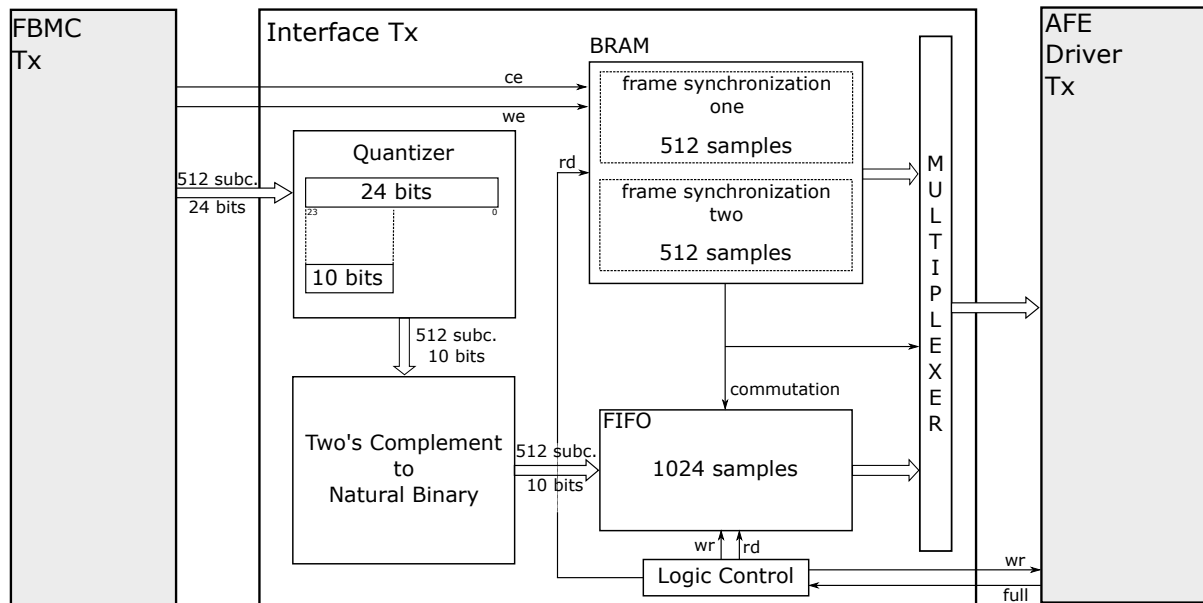


Figura 7.2: Esquema del módulo Interfaz en transmisión.

sus características de transmisión están configuradas para una frecuencia de transmisión de 61.5 MHz con un ancho de banda de 31.25 MHz, con una frecuencia de trabajo de 100 MHz. Debido a las limitaciones del AFE explicadas en el capítulo 4, el ancho de banda máximo que se puede transmitir corresponde con 25 MHz. Es por tanto necesario introducir un retardo superior al establecido para conseguir los 50 MHz en la transmisión para pasar de 31.25 MHz a 25 MHz y ajustarse al requerido. En la figura 7.3 se puede ver cómo se consigue cambiar pasar de una transmisión de 100 MHz a 50 MHz mediante flags *ce* y *we* de habilitación.

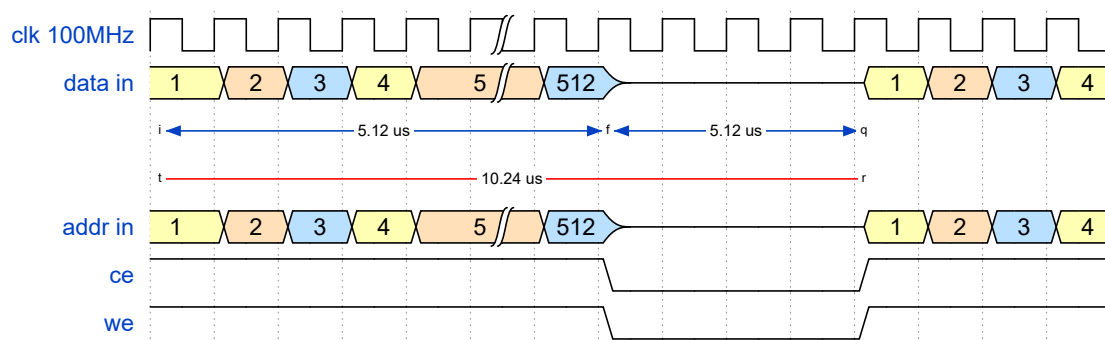


Figura 7.3: Cronograma para transmitir las tramas a 50 MHz con un reloj de 100 MHz.

Tanto la memoria BRAM como la memoria FIFO se encuentran situadas en el dominio de reloj de los 100 MHz, pero deben funcionar a 50 MHz para adecuarse al funcionamiento del AFE. Tanto la lectura de las muestras de las tramas de sincronismo como la escritura y lectura de las subportadoras del bloque FBMC transmisor se hará a esta frecuencia; es por tanto necesario implementar la lógica necesaria para adecuar la frecuencia de funcionamiento con la de transmisión de los datos. Como la frecuencia de funcionamiento es de 100 MHz y se requiere trabajar a 50 MHz, se debe trabajar cada dos ciclos de 100

MHz, esto equivaldría a hacerlo a 50 MHz. En el cronograma de la figura 7.4 se puede apreciar la señal que habilita la lectura cada dos ciclos como *contador BRAM*.

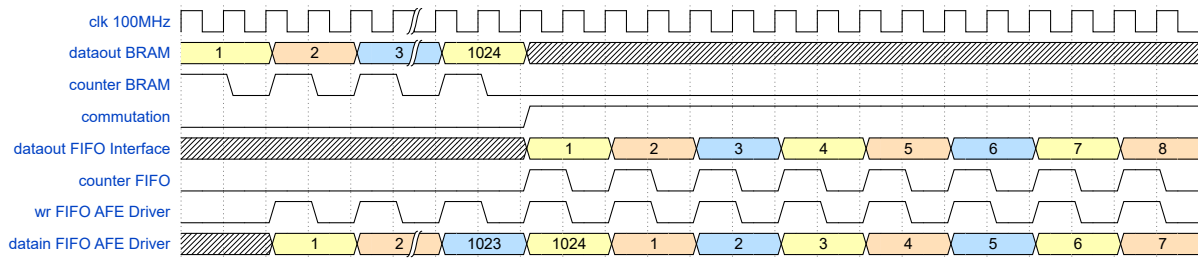


Figura 7.4: Cronograma del funcionamiento de la conmutación entre memorias.

La lógica para controlar ambas memorias será la siguiente. Se inicia transmitiendo las tramas de sincronismo, se transmiten cada dos ciclos de 100 MHz, lo que equivale a 50 MHz. Cuando se activa la lectura, además de leer se aumenta la dirección de lectura de la BRAM para la siguiente y se habilita la escritura de la memoria FIFO de transmisión del *AFE Driver* transmisor. Una vez se han transmitido las 1024 muestras de sincronismo, se conmuta a la memoria FIFO de la interfaz y mediante un multiplexor se selecciona la salida de la FIFO para transmitir al *AFE Driver* transmisor. La lógica de la memoria FIFO es la misma que la de BRAM, cada dos ciclos se lee la memoria. La figura 7.4 representa el cronograma de todo lo explicado en este párrafo, la lógica para pasar de 100 MHz a 50 MHz, la conmutación y la escritura en el AFE Driver.

El control de la memoria FIFO del *AFE Driver* transmisor que transmite las muestras hacia el AFE se realiza en el módulo *Interfaz*. Se basa en el estado del flag *full* de la FIFO. En este caso, al producirse y consumirse datos a la misma frecuencia, no habrá desbordamientos de la memoria, pero se establece la lógica necesario en caso de que no existiera dicha

### 7.3 Interfaz Receptor

Una vez se trasmiten las tramas de sincronismo y las subportadoras de las tramas de información desde la interfaz hacia el *AFE Driver* transmisor, se envían por el canal PLC mediante la conversión DA del AFE. Se reciben en recepción por el AFE y se almacenan en la memoria FIFO del módulo *AFE Driver* a 50 MHz. La frecuencia de lectura de esas muestras es 100 MHz, por lo que es necesario implementar una lógica de control. Ésta se basa en el flag *empty* de la memoria FIFO de recepción del *AFE Driver*. Se trata de leer únicamente cuando la memoria desactive este flag a 100 MHz, pero no llega a vaciarla del todo, por lo que va leyendo a una frecuencia media de 50 MHz. En la figura 7.5 se puede ver el proceso de lectura con su control correspondiente.

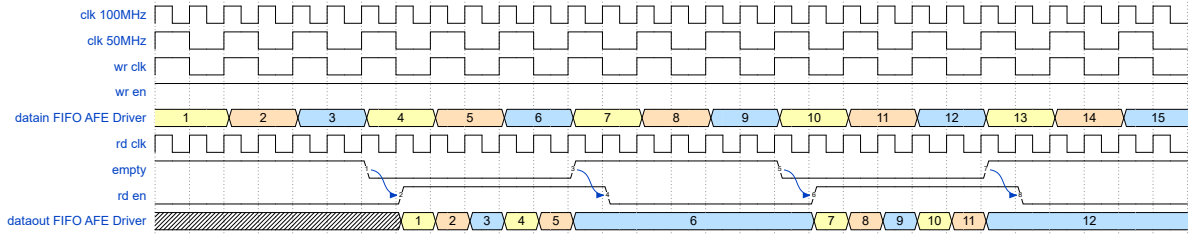


Figura 7.5: Cronograma de la lectura de la FIFO del *AFE Driver*.

Las muestras que se leen de la FIFO receptora del *AFE Driver* receptor se escriben en paralelo en una memoria *ping pong* y en una memoria FIFO que se encuentran en el interior del módulo *Interfaz*. La memoria *ping pong* consta de 4 memorias de doble puerto de 512 muestras cada una, que funcionan dos-a-dos para proporcionar bloques de 1024 muestras al módulo de *Sincronismo*. Se requieren dos memorias de 512 muestras para proporcionar 1024 debido a la partición del bus de entrada al sincronismo. De manera conceptual, es una memoria *ping pong* con dos bloques de 1024 posiciones, uno transmite las 1024 muestras almacenadas y el otro se encuentra de backup mientras las transmite al módulo *Sincronismo*. En la figura 7.6 se puede ver el procedimiento de escritura de las memorias de doble puerto y en las figuras 7.7 y 7.8 se puede ver el proceso de transmisión al bloque de sincronismo. La memoria FIFO tiene como finalidad almacenar todas las muestras recibidas mientras se comprueban en el módulo de *Sincronismo*, ya que si se encuentra el inicio de la primera trama en las muestras que se han introducido al bloque de sincronismo, es necesario luego recuperarla a través de la posición que devuelva dicho módulo.

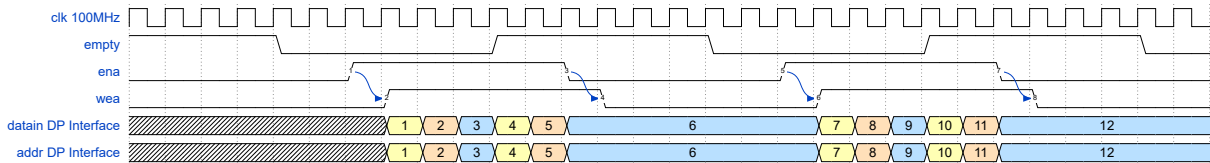


Figura 7.6: Cronograma de la escritura de las memorias DUAL PORT.

Esta memoria FIFO en la que se almacenan las muestras en paralelo tiene que tener un tamaño ( $C_{FIFO}$ ) de como mínimo que satisfaga la ecuación 7.1.

$$C_{FIFO} \geq t_D + t_S \quad (7.1)$$

La latencia de datos ( $t_D$ ), medida en número de ciclos de reloj de 50 MHz, corresponde con el tiempo de llenado de las 2 memorias de doble puerto, que tiene una latencia de 2048 ciclos de 100 MHz (1024 ciclos a 50 MHz); y la latencia del módulo de sincronismo ( $t_S$ ) que corresponde con la latencia del módulo para realizar la comprobación, también medida en número de ciclos de reloj, que son 10339 ciclos de 100 MHz (5170 ciclos a 50 MHz).



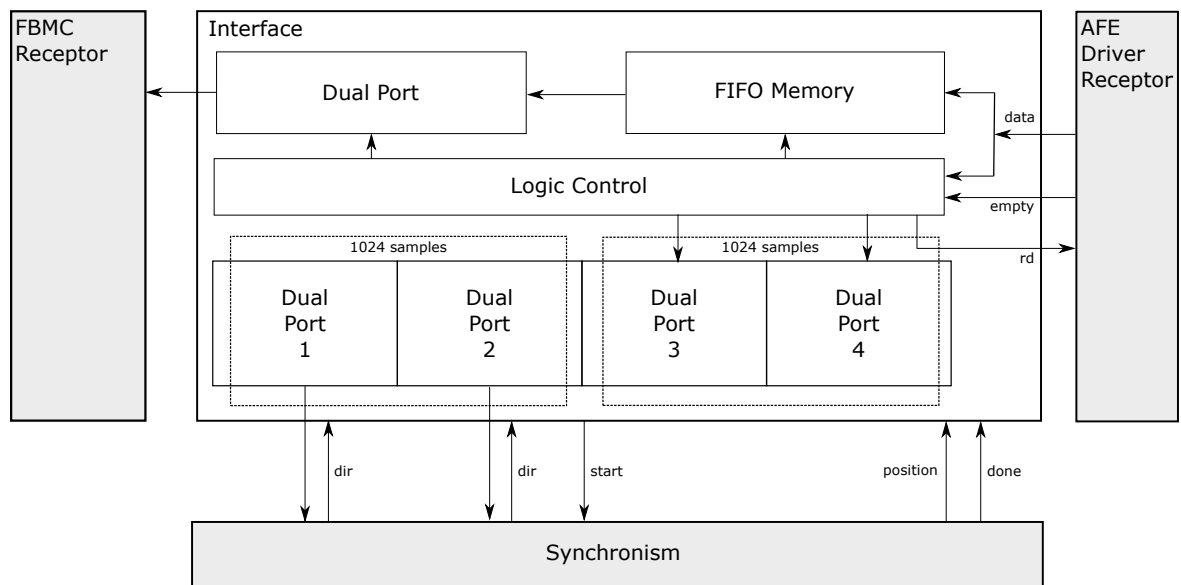


Figura 7.7: Funcionamiento de la conmutación entre memorias de doble puerto.

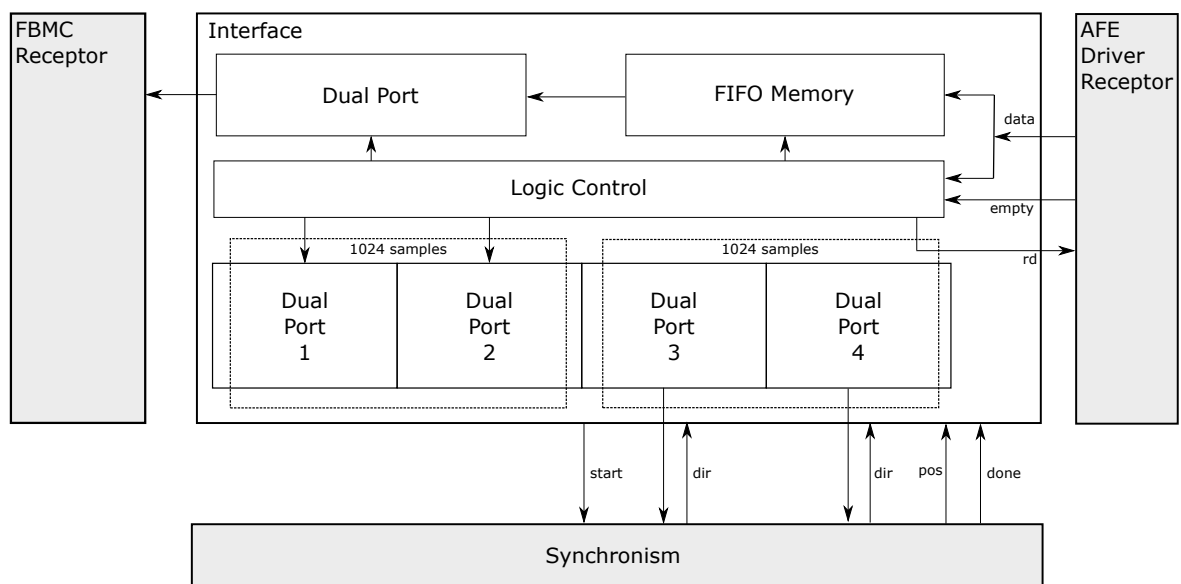


Figura 7.8: Funcionamiento de la conmutación entre memorias de doble puerto.

$$C_{FIFO} \geq 1024 \text{ ciclos} + 5170 \text{ ciclos} = 6194 \text{ ciclos}$$

La latencia total es de 6194 ciclos de 50 MHz (frecuencia a la que se reciben datos), por tanto, la memoria FIFO debe de tener un múltiplo de 2 superior de tamaño, correspondiendo con 8192 posiciones.

El bloque de sincronismo entrega el flag *done* cuando termina de evaluar las 1024 muestras tras 10339 ciclos de latencia de 100 MHz, con cada activación del flag *done* se evalúa una sección de 1024 posiciones de la memoria FIFO, que contiene las mismas muestras que evalúa el sincronismo. Si el módulo de *Sincronismo* detecta un valor por encima del umbral en ese bloque de muestras, se detiene y se busca; en caso contrario, se sigue realizando la comprobación del sincronismo y después se evalúa el siguiente sector. En la figura 7.9 se puede ver esa evaluación por bloques de 1024 muestras.

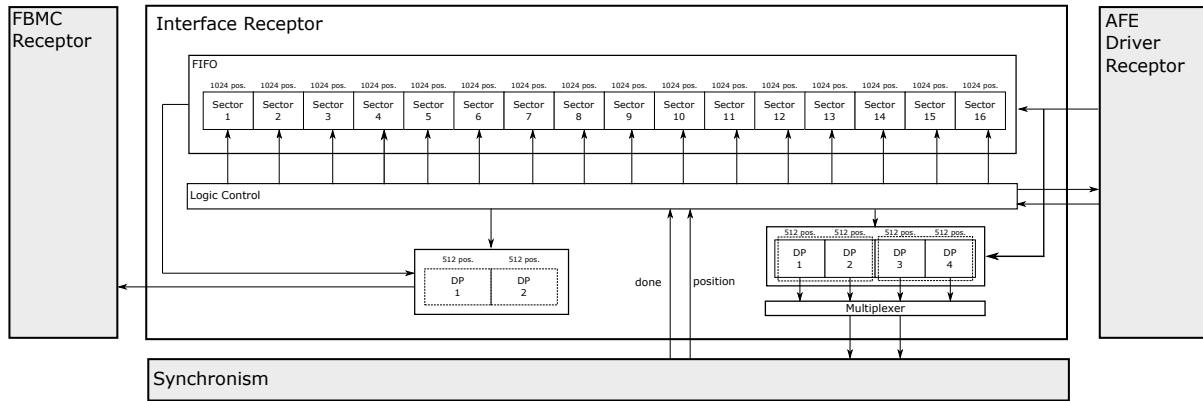


Figura 7.9: Composición en sectores de la memoria FIFO que almacena las muestras recibidas mientras se busca la primera trama de información.

Una vez se encuentra la primera subportadora de la primera trama, se transmiten ordenadas desde la FIFO a dos memorias de doble puerto de 512 posiciones cada una. Se establecen dos debido a que una vez se escribe una y se procede a leerla, se deben seguir escribiendo, por lo que se establece una de backup, y 512 debido a que es el tamaño de la trama transmitida por el FBMC transmisor. En la figura 7.10 se representa esta transmisión final desde la interfaz al receptor FBMC.

En la figura 7.11 se puede ver cómo se realiza este proceso de escritura y lectura. Además, en la interfaz de transmisión se redujo el tamaño de palabra a 10 bits para ajustarlo al tamaño del bus del AFE; ahora es necesario realizar una cuantificación y devolverlo a su tamaño original de 24 bits. Los 10 bits se pondrán como los de mayor peso y el resto se rellenarán con ceros. La lectura de las tramas se realiza de forma inversa por requerimientos del FBMC receptor.

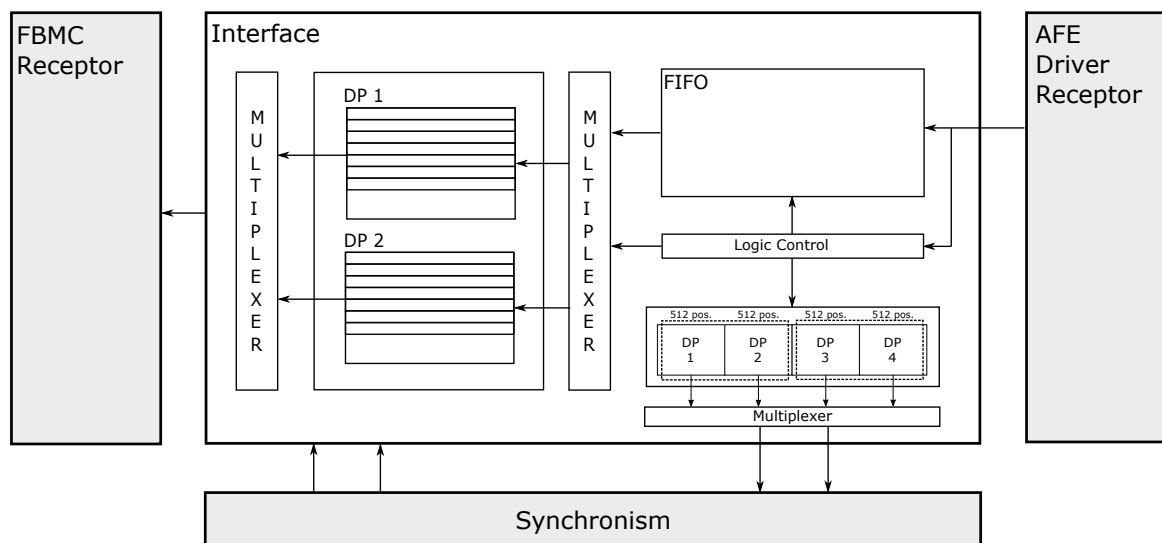


Figura 7.10: Envío de tramas al FBMC receptor desde la interfaz.

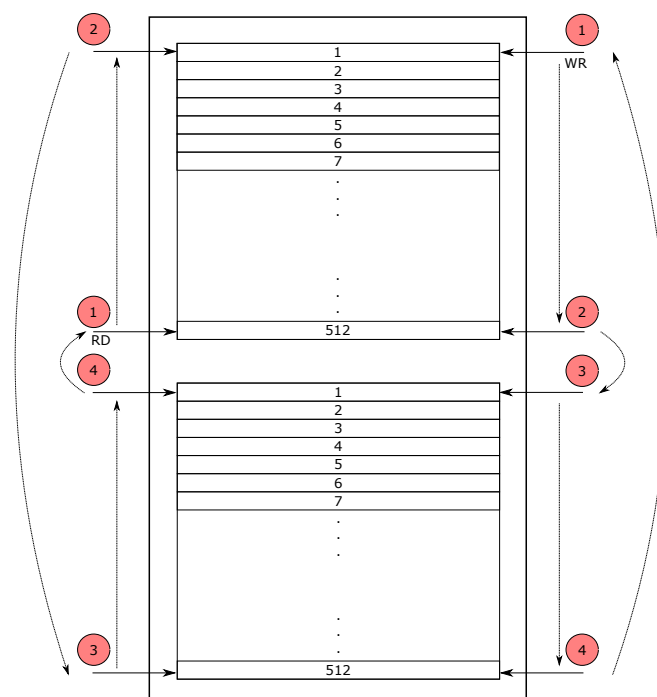


Figura 7.11: Escritura y lectura de la memoria DUAL PORT que proporciona las tramas al FBMC receptor.



## Capítulo 8

# Resultados

### 8.1 Introducción

En este capítulo se presentan los resultados parciales y finales del trabajo junto con las simulaciones experimentales oportunas. A lo largo de los diferentes capítulos presentados, se ha realizado la descripción de cada módulo y su función dentro del diseño de la arquitectura planteada. Ahora, se integran todos ellos para presentar resultados en conjunto y comprobar que todo funciona según los planteamientos teóricos previos y las simulaciones parciales.

En un primer apartado, se comentarán los recursos consumidos por el diseño y la ocupación en función del SoC utilizado. En el siguiente punto, se mostrarán simulaciones globales de todo el diseño donde se prueba que la arquitectura funciona según lo esperado. Y por último, se procede a presentar capturas experimentales obtenidas mediante la implementación en placa de todo el diseño.

Para obtener estos resultados será necesario hacer uso de varias herramientas. Para obtener las simulaciones globales del diseño se utiliza el simulador de la herramienta de desarrollo. Para las capturas experimentales es necesario dividir las señales que salen de la tarjeta de las internas dentro de la arquitectura. Para las capturas de las señales externas a la tarjeta se hará uso de un osciloscopio digital, y para las capturas internas se hará uso del CORE de Xilinx ILA.

### 8.2 Recursos utilizados

La arquitectura ha sido integrada como un periférico avanzado en un SoC basado en la familia Zynq, de Xilinx, Inc. Concretamente se ha implementado sobre el dispositivo ZC706. En la tabla [8.1](#) se pueden ver los recursos consumidos por la arquitectura que se plantea

en este trabajo, incluyendo los módulos que se han descrito a lo largo de los capítulos, como son: el transmultiplexor FBMC, la interfaz, los drivers del AFE y el sincronismo, así como los disponibles en el dispositivo. En esas condiciones de implementación, el sistema ha sido sintetizado para una frecuencia de reloj global de 100 MHz.

Tabla 8.1: Recursos consumidos por la arquitectura propuesta.

	Utilización	Disponible	Utilización %
<b>Look-up Tables (LUT)</b>	71584	218600	32.74
<b>RAM built for Look-Up Tables (LUTRAM)</b>	35726	70400	50.74
<b>Flip-Flop (FF)</b>	25062	437200	04.48
<b>Block Random Access Memory (BRAM)</b>	193	545	35.41
<b>Digital Signal Processor (DSP)</b>	146	900	16.22
<b>Inputs and Outputs (IO)</b>	42	362	11.60
<b>Global Clock Buffer (BUFG)</b>	9	32	28.12
<b>Mixed-Mode Clock Manager (MMCM)</b>	1	8	12.50

Se ha podido ver en la tabla que el consumo de recursos medio está en torno al 24 %. Aunque no parezca una cifra muy elevada, es una cifra considerable por la cantidad de recursos disponibles, además de alcanzarse en algunos casos hasta el 50 % de recursos consumidos.

### 8.3 Simulaciones globales

Para verificar la arquitectura propuesta, se realiza un modelo sin incluir el transmultiplexor FBMC: esto supone un diseño con la interfaz, los dos drivers y el sincronismo. De esta manera, se comprueba que los módulos realizados y simulados por separado funcionan en conjunto de la manera deseada. Con estos módulos, se deben comprobar los datos a la entrada y salida de la interfaz transmisora, del driver transmisor, del driver receptor y de la interfaz receptora cuando el módulo de *Sincronismo* determine la primera subportadora de la primera trama transmitida.

A la entrada de la interfaz, se sustituye el transmisor FBMC por una entrada conocida generada en el banco de pruebas. Se opta por introducir valores en el rango  $[0, 511]$ . La explicación sobre el establecimiento de esta entrada, radica en el hecho de numerar las 512 subportadoras de cada trama, y de esta manera mantenerlas identificadas a lo largo de todo el proceso de comunicación. Se reemplaza el transmultiplexor FBMC, pero se busca simular su comportamiento para realizar simulaciones lo más fieles posibles a la realidad. Por tanto, se generan 512 datos a 100 MHz, lo que equivale a una transmisión de  $5.12 \mu\text{s}$ , y se mantiene sin transmitir otros  $5.12 \mu\text{s}$ . Esto provoca que la transmisión media sea a una frecuencia de 50 MHz con un ancho de banda de 25 MHz, el máximo del AFE utilizado. Esta transmisión se realiza de esta manera para simular el comportamiento del transmisor FBMC explicado en el capítulo 7. En la figura 8.1 se puede ver la entrada a la interfaz transmisora, con periodos de  $5.12 \mu\text{s}$  de transmisión y  $5.12 \mu\text{s}$  de espera.

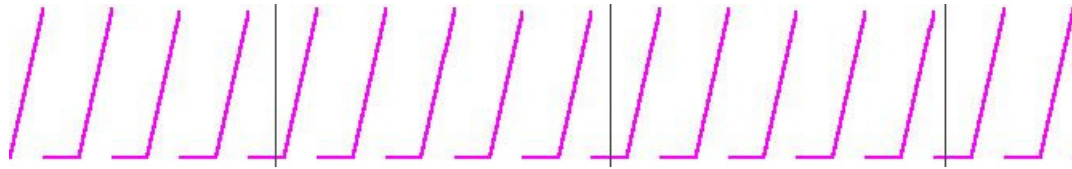


Figura 8.1: Valores de entrada al sistema simulando el transmisor FBMC.

Una vez se comienza la transmisión de datos a la entrada de la interfaz transmisora, se habilita la transmisión de las tramas de sincronismo que se encuentran almacenadas en una memoria BRAM dentro de la interfaz transmisora. Mientras tanto, se almacenan los datos en una memoria FIFO hasta que se envían las dos tramas de sincronismo, momento en el cual comienza la transmisión de éstas. Por tanto, a la salida se debe obtener las dos tramas de sincronismo concatenadas con las tramas de datos. Además, deben de transmitirse a una frecuencia de 50 MHz pese a estar en el dominio de reloj de los 100 MHz, esto se realiza mediante una lógica de control de estas memorias. En la figura 8.2 se puede ver cómo se cumple esta descripción y se distinguen perfectamente un tipo de trama de la otra.

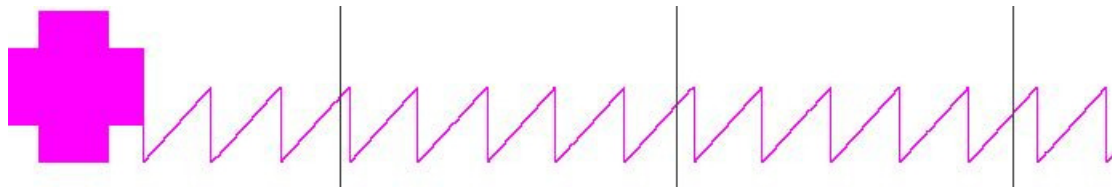


Figura 8.2: Tramas de sincronismo y de datos a la salida del módulo *Interfaz transmisor*.

Una vez salen del módulo *Interfaz*, se dirigen hacia el driver del AFE transmisor. Como se comentó en el capítulo 5, una de sus funciones en el diseño radica en la realización del cambio de dominio de reloj de 100 MHz a 50 MHz. Como se adapta la transmisión a la entrada del driver, no se hará uso de la lógica de control de la transmisión mediante su memoria FIFO interna, pero queda implementado para poder establecer anchos de banda mayores. Por tanto, a la salida del driver se debe obtener lo mismo que a la entrada, como se puede ver en la figura 8.3 en comparación con la figura 8.2.

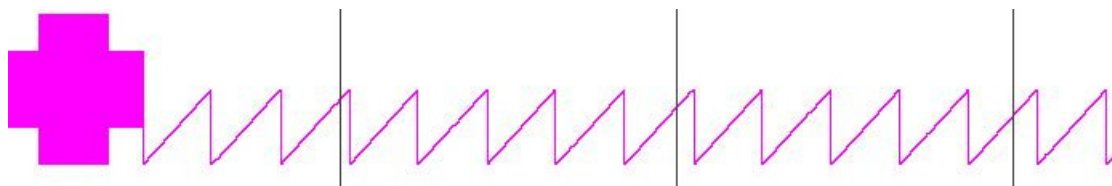


Figura 8.3: Salida del driver transmisor del AFE.

Al ser una comprobación de la arquitectura del diseño, se realiza la conexión interna dentro del dispositivo FPGA de los drivers del AFE simulando un canal PLC ideal, por tanto la salida del driver transmisor será la propia entrada del receptor. En este driver, se debe realizar el cambio de dominio de reloj inverso al realizado en transmisión, es decir, de 50 MHz a 100 MHz. Al consumirse los datos a una frecuencia superior de la que se producen,

como ya se comentó en el capítulo 7, es necesario establecer una lógica de control basada en el flag *empty* de la memoria FIFO interna. Por tanto, a la salida se obtendrán los datos recibidos del canal ideal pero a una frecuencia de 100 MHz con periodos de inactividad hasta que se deshabilita el flag. En la figura 8.4 se puede ver la salida del driver receptor, coincidiendo con lo visto en las anteriores 8.2 y 8.3.

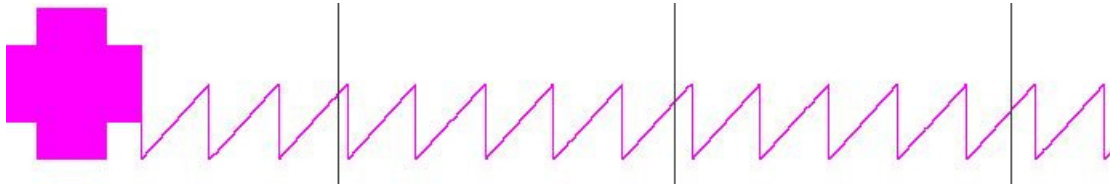


Figura 8.4: Salida de las tramas del driver receptor con un canal PLC ideal.

Una vez se obtienen 1024 muestras desde el driver receptor, se introducen en paralelo en el módulo de *Sincronismo* (previo paso por una memoria *ping pong*) y en un sector de la memoria FIFO de la interfaz receptora, como se vio en el capítulo 7. En la figura 8.5 se puede ver la entrada al módulo de *Sincronismo*, cuyas 1024 primeras muestras introducidas incluyen una trama de sincronismo completa. La entrada se desglosa en dos buses de 512 muestras como se puede ver en la figura.

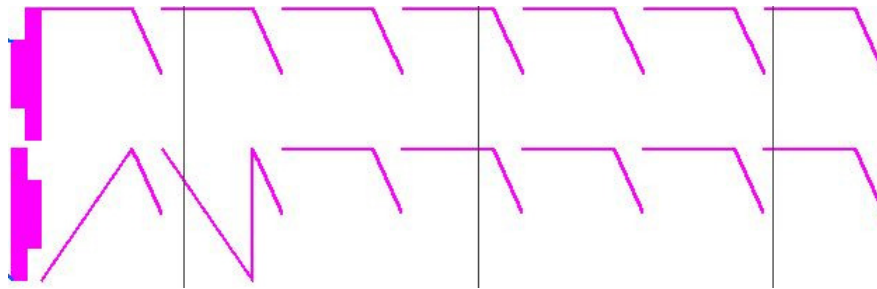


Figura 8.5: Bloques de 1024 muestras introducidos al módulo de *Sincronismo*.

Al introducirse una trama de sincronismo completa en el módulo de *Sincronismo*, se produce un pico de correlación con el cual, se puede determinar la primera subportadora de la primera trama de datos. En la figura 8.6 se representa el resultado de la correlación cruzada en el dominio del tiempo con el pico mencionado.

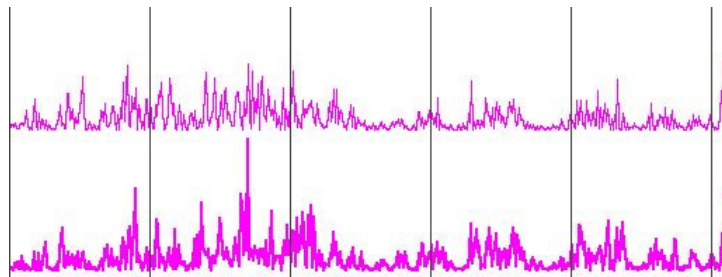


Figura 8.6: Pico de correlación obtenido del módulo de *Sincronismo*.

Con la posición del pico de correlación, se determina cuál es el inicio de la trama de datos en el bloque de 1024 muestras introducido. Como ya se comentó en el capítulo 7, la división



en bloques de 1024 que se realiza para introducir en el módulo de *Sincronismo*, se realiza también sectorizando la memoria FIFO, por lo que, cuando se determina el inicio de los datos en el módulo de *Sincronismo*, equivale a obtenerlo en la memoria FIFO. Una vez se conoce la posición dentro de la FIFO que contiene la primera subportadora, se envían a una memoria *ping pong* para transmitir las al receptor a 100 MHz y de manera inversa para ajustarse a los requerimientos de la interfaz *ap memory* existente en el receptor FBMC. Realizando correctamente todo el proceso, se deben obtener las tramas transmitidas de manera inversa y a 100 MHz, como se puede ver en la figura 8.7.

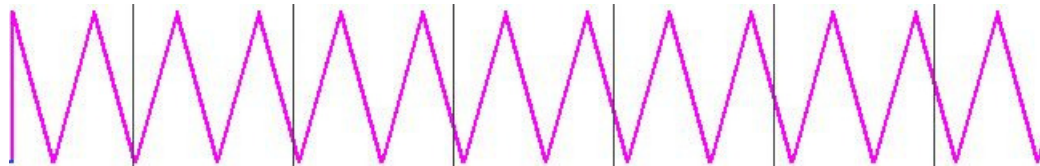


Figura 8.7: Salida del módulo *Interfaz* de recepción.

Validada la arquitectura de manera individualizada, se incorporan a ella tanto el transmultiplexor FBMC como los AFE utilizados, para realizar el montaje real. También se incorpora el canal PLC para la transmisión de la información.

## 8.4 Capturas experimentales

Como se ha comentado, en este apartado se utiliza todo el diseño completo para comprobar su funcionamiento, contando con: el transmultiplexor FBMC, la interfaz, los dos drivers juntos con su AFE correspondiente y el módulo de *Sincronismo*. Se sintetiza y se implementa en la tarjeta utilizada. Para obtener resultados internos, se implementará el CORE de Xilinx ILA, que se utiliza como una sonda interna dentro de la FPGA; y para las señales externas a la tarjeta un osciloscopio digital.

A la entrada del módulo transmisor FBMC se establece una entrada conocida (una señal senoidal en este caso); de esta manera, cuando se reciba correctamente y se introduzca en el receptor FBMC, se obtendrá de nuevo la señal identificándola correctamente.

Tras pasar por el banco de filtros del transmisor FBMC, la señal se convierte en información no interpretable visualmente, como si pasaba inicialmente con la entrada senoidal, y se transmite hacia el módulo *Interfaz* de transmisión. En la figura 8.8 se puede ver la salida del transmisor FBMC, tramas de 512 subportadoras cada una a 100 MHz durante  $5.12 \mu s$  y otros  $5.12 \mu s$  de espera.

Mediante el CORE ILA se obtiene la salida de la interfaz, en donde al principio de la transmisión, se deben encontrar dos tramas de sincronismo y a continuación las tramas



Figura 8.8: Captura experimental de la salida del transmisor FBMC.

de datos, como ya se vio en el apartado previo para este mismo tramo. En la figura 8.9 se puede ver dicha salida y la distinción entre tipos de tramas.

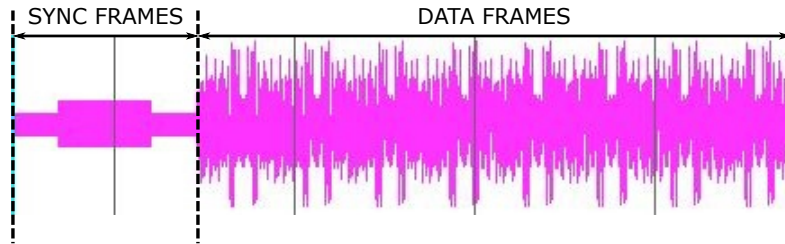


Figura 8.9: Captura experimental de las tramas de sincronismo y de datos.

Dicha salida se introduce en el driver del AFE transmisor. A la salida del mismo, se envía hacia el AFE, encargado de realizar la conversión DA y mandar hacia el canal PLC. Mediante el osciloscopio digital se pueden ver estas tramas enviadas al canal. En la figura 8.10 se pueden ver las dos tramas de sincronismo transmitidas. Estas tramas tienen 1024 muestras, transmitidas a 50 MHz, es decir, una muestra cada 20 us, el tiempo necesario sería de 20.48 us. En la parte inferior de la figura 8.10, en  $(\Delta t)$ , se puede comprobar que coincide.

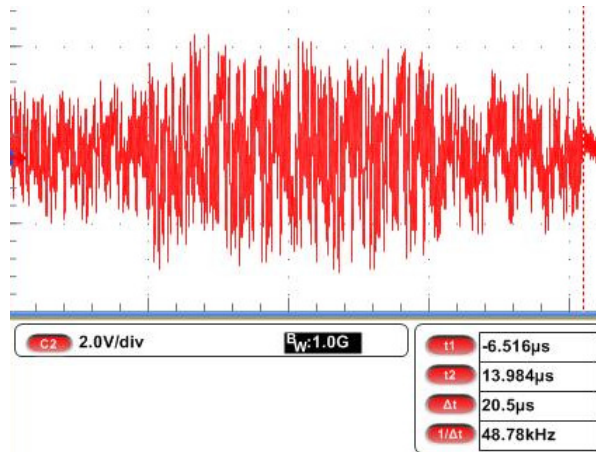


Figura 8.10: Captura experimental de las tramas de sincronismo transmitidas al canal mediante el AFE transmisor.

Mediante la obtención del espectro de la señal transmitida con el osciloscopio, se puede ver en la figura 8.11 cómo toda la energía se encuentra en los 25 MHz ( $\Delta t$ ). Esto coincide con lo establecido inicialmente en el diseño, transmitiendo con un ancho de banda de 25 MHz, lo máximo posible para el AFE utilizado.

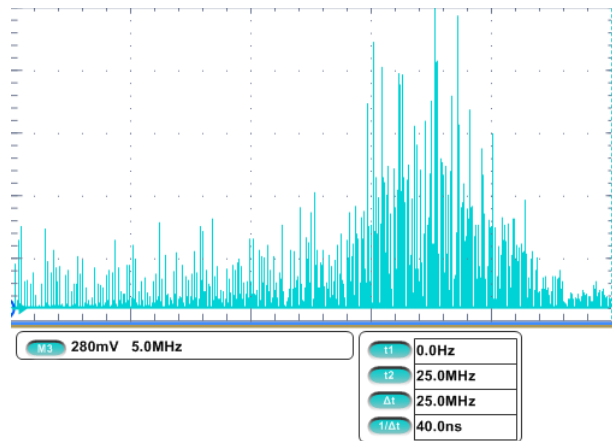


Figura 8.11: Captura experimental del espectro de la señal transmitida por el AFE al canal PLC.

A la salida del driver receptor mediante el CORE ILA, se puede ver cómo la trama se recibe atendiendo a la lógica establecida para la lectura mediante el flag *empty*. Esto provoca periodos de lectura continuada y periodos de espera hasta la próxima. En la figura 8.12 se puede ver la salida del driver receptor hacia la interfaz comparada con la transmitida a la entrada del driver. Como cabía esperar, se puede ver, la señal se deteriora debido al efecto del canal PLC existente.

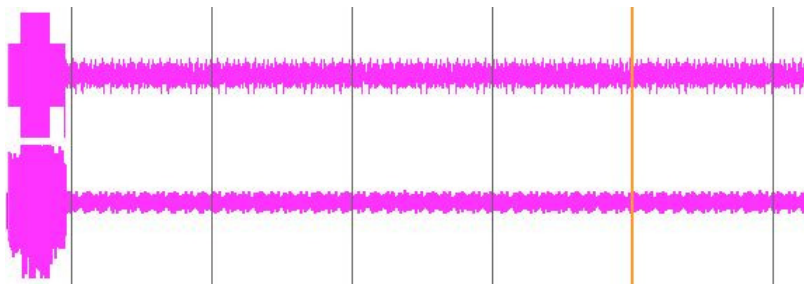


Figura 8.12: Captura experimental de la entrada del driver transmisor y de la salida del driver receptor, tras pasar por el canal PLC.

A la entrada del módulo de *Sincronismo*, se introducen las muestras recibidas, obteniéndose lo representado en la figura 8.13, donde se puede ver como se desglosa el bloque de 1024 muestras en dos de 512, y que además entra una trama de sincronismo completa, provocando el pico de correlación correspondiente.

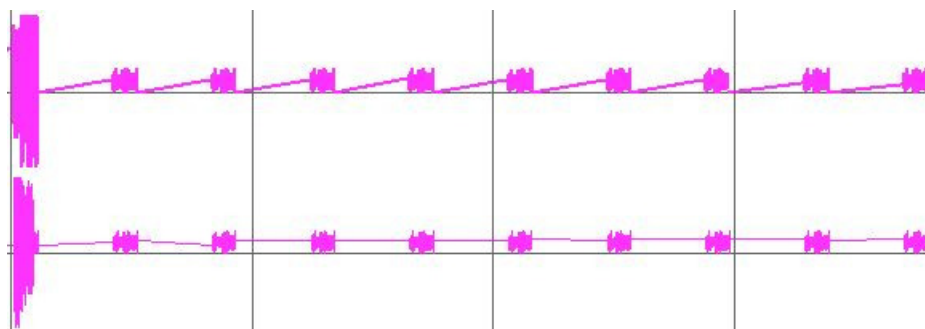


Figura 8.13: Captura experimental de la entrada al módulo de *Sincronismo*.

A la salida del módulo de *Sincronismo*, si se ha introducido una trama de sincronismo completa, como es este caso, se debe ver el pico de correlación. En la figura 8.14 se puede observar claramente varios picos pronunciados. Como ya se comentó en el capítulo 6, se realiza la correlación cruzada mediante un producto en el dominio de la frecuencia de: la transformada de las tramas recibidas desde el canal PLC y las tramas de sincronismo ideales. Para pasar del dominio temporal al dominio transformado es necesario realizar una FFT. La arquitectura escogida y explicada también en el capítulo 6, es idónea para su implementación hardware pero presenta el inconveniente de obtener las muestras desordenadas. Por ello, el pico de correlación que presenta unos lóbulos laterales en posiciones contiguas, en esta figura, los picos presentes salen separados unos de otros, pero se toma la posición del de mayor amplitud.

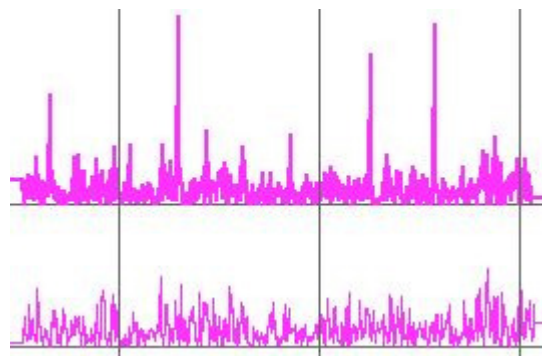


Figura 8.14: Captura experimental de la salida del módulo de *Sincronismo* con el pico de correlación correspondiente.

Una vez el pico de correlación determina con su posición el inicio de las tramas de datos dentro del bloque de 1024 muestras introducido, se busca el comienzo en la memoria FIFO de la interfaz y se transmite hacia el receptor FBMC, pasando previamente por una memoria *ping pong* para transmitir las a 100 MHz. A la salida de la interfaz, se debe obtener la trama de datos transmitida que se vio en la figura 8.8. En la figura 8.15 se puede ver una comparativa entre la recibida (parte superior) y la transmitida (parte inferior).



Figura 8.15: Captura experimental de la comparativa entre la señal recibida y la transmitida.

A la salida del receptor FBMC se puede ver la información que se transmitió en primera instancia. En la figura 8.16 se puede ver la señal recibida tras pasar por el receptor FBMC, coincidiendo con la señal senoidal transmitida. Al no utilizarse técnicas de estimación e igualación de canal, la señal se encuentra deteriorada comparada con la transmitida.



Figura 8.16: Captura experimental de la salida del receptor FBMC.

Se procede ahora a mostrar los dos montajes posibles, realizando todo el diseño en un único SoC, siendo el transmisor y receptor, como se puede ver en la figura 8.17 o en la figura 8.18, donde se puede ver el montaje mediante dos SoC, dividiendo la arquitectura en un bloque transmisor implementado en un SoC, y el bloque receptor implementado en el SoC restante.



Figura 8.17: Montaje realizado mediante un único SoC.





Figura 8.18: Montaje realizado mediante dos SoC.

## Capítulo 9

# Conclusiones y líneas futuras

En este capítulo se ofrecen las conclusiones obtenidas en el desarrollo de este trabajo y se proponen futuras líneas en las que investigar a partir de las mismas.

### 9.1 Conclusiones

Este trabajo ha permitido la definición y diseño de una arquitectura sobre una FPGA para el desarrollo de una comunicación PLC, a partir del desarrollo de una interfaz, un driver y un sincronismo entre un trasmultiplexor FBMC y el correspondiente módulo AFE. A partir de aquí, se pueden extraer una serie de conclusiones parciales:

- Se han podido realizar comunicaciones PLC de banda ancha y alta velocidad, al tener anchos de banda de 25 MHz (el máximo que permite el AFE utilizado) y hasta 50 Msps.
- Este canal está presente en todo tipo de aeronaves y vehículos, por lo que ofrece un medio de comunicación de banda ancha y evita la implementación de otra red paralela, ya que la red eléctrica se encuentra implementada para la alimentación. Las futuras aplicaciones que se propongan tendrán un ancho de banda considerable debido al aumento de la complejidad y requerirán de mucho ancho de banda.
- Se utiliza la técnica de acceso al medio recomendada por el estándar de comunicaciones PLC de banda ancha IEEE 1901-2010 como es FBMC.
- Se implementa un mecanismo de sincronismo que ofrece muy buenos resultados para comunicaciones PLC, al identificarse claramente el pico de correlación con el que determinar el inicio de la comunicación.
- Para las etapas de conversión, integración, ganancia y transformación se ha hecho uso de los Analog Front-End, que incorporan en un único dispositivo todos los elementos requeridos.

- Se ha conseguido manejar el AFE utilizado mediante un driver, que permite configurarlo como transmisor o receptor a través del manejo de las señales enviadas a sus terminales. También realiza el cambio de las características de funcionamiento y el tratamiento de los datos.
- Para la implementación de la propuesta se ha hecho uso de un SoC, en el cual se ha integrado el diseño de la arquitectura planteado en la PL, y se gestiona desde la PS.
- Los recursos requeridos para implementar la arquitectura son bastante elevados pero no superan los máximos de la tarjeta utilizada.
- Se ha validado la propuesta mediante pruebas experimentales, donde se ha podido comprobar el funcionamiento del sistema planteado.

## 9.2 Líneas futuras

Se presentan a continuación las posibles líneas futuras desde las cuales poder utilizar el trabajo de base.

- Para obtener mejores resultados, la inclusión de la estimación e igualación de canal en la propuesta es un apartado importante a añadir para continuar el desarrollo.
- Para poder ajustarse a los 31.25 MHz de ancho de banda del estándar IEEE 1901-2010, se requieren AFE con conversores AD/DA que tengan mayores tasas de conversión.
- Para utilizar la arquitectura y cumplir las especificaciones de un estándar de comunicaciones PLC de banda ancha, emplear el estándar IEEE 1901.1-2018, que está pensado para un ancho de banda de 12 MHz y es inferior a los 25 MHz obtenidos.
- Al tener el SoC varios conectores FMC, se podrían conectar varios AFE transmitiendo en un SoC y en otro varios recibiendo, estableciendo las técnicas MIMO para aumentar el régimen binario y reducir la probabilidad de error de la transmisión.
- La realización de medidas con diferentes características del canal PLC para la obtención de parámetros característicos que definen la calidad de una comunicación.



# Capítulo 10

## Pliego de condiciones

En este capítulo se detallarán las herramientas que se han utilizado para realizar este trabajo, tanto hardware como software.

- Elementos Hardware
  - *Ordenador utilizado*
    - \* Procesador: Intel Core<sup>TM</sup> i7 CPU, 2.93 GHz
    - \* Memoria RAM: 4 GB
  - *Módulo AFE basado en el MAX2981*
    - \* Fabricante: Maxim Integrated<sup>TM</sup>
    - \* Tamaño de palabra de los conversores: 10 bits
    - \* Frecuencia de conversión: 50 Msps
  - *SoC ZC706*
    - \* Fabricante del dispositivo FPGA: Xilinx
    - \* Número de bits: 32
    - \* Familia del dispositivo FPGA: Zynq-7000
    - \* Arquitectura del procesador: ARM
    - \* Subarquitectura del Núcleo: Cortex<sup>TM</sup>-A9
- Elementos Software
  - *Sistemas operativos utilizados*
    - \* Windows 10 Pro
  - *Software para el desarrollo del trabajo*
    - \* Matlab R2018b
    - \* Vivado

- \* Vivado HLS
- \* Xilinx Software Development Kit (SDK)
- \* Questa
- *Software para el tratamiento de textos*
  - \* Texpad 1.8.7
- *Software para el tratamiento de imágenes*
  - \* Inkscape
  - \* GIMP 2

## Capítulo 11

# Presupuesto

En este capítulo se presentan los diferentes costes de todo el trabajo incluyendo: costes materiales, mano de obra, gastos generales y beneficio industrial. Existen dos tipos de coste: Coste Directo (CD) e Coste Indirecto (CI). El coste directo engloba el coste material y la mano de obra, y el coste indirecto corresponde con los gastos generales.

En la tabla 11.1 se presentan los gastos referentes a los elementos de material y software utilizado, considerado un CD. El valor de los equipos electrónicos tiene una amortización estimada de 4 años, por lo que se debe hacer una relación en función de los meses que se han utilizado.

Tabla 11.1: Coste material utilizado en el trabajo (CD).

Desglose de costes en términos de material			
Material/Software	Precio (euros)	Utilización/Amortización	Coste al proyecto (euros)
Ordenador	1000	8 meses/4 años	166.66
Windows 10	40	8 meses	40
Licencias Xilinx	3123	8 meses	3123
Licencia Matlab	35	8 meses	35
Microsoft Office	70	8 meses	70
Latex	0	8 meses	0
ZC706	2237	8 meses/4 años	372.83
2 AFE	400	8 meses	400
Material Ofimático	50	8 meses	50
InkScape	0	8 meses	0
Gimp	0	8 meses	0

El coste total de material y software empleado para la realización de este trabajo es de 4257.49 euros sin incluir impuestos.

En la tabla 11.2 se muestra el coste por mano de obra del trabajo. Se representa por separado el precio por horas de ingeniería y de redacción del trabajo, cuyo coste será diferente en función de la dificultad de cada trabajo. Este coste se considera un coste directo también al igual que el coste de material.

Tabla 11.2: Coste de mano de obra en el trabajo (CD).

Desglose de costes en términos del trabajo realizado			
Trabajo	Número de horas	Precio por hora (euros)	Total (euros)
Ingeniero	400	60	24000
Redacción	150	20	3000

El coste total por mano de obra es de 27000 euros sin incluir IVA, que corresponde con la suma del coste por trabajo de ingeniería y de redacción.

El total de los costes directos (CD) corresponde con la suma de los costes de material (4257.49 euros) y de mano de obra (27000 euros), por lo que los costes directos del trabajo son 31257.49 euros.

Se incluyen los Gastos Generales al proyecto, considerados costes indirectos (CI), que se calculan como el 15 % del CD. También se suma el Beneficio Industrial que se calcula como el 6 % de la suma del CD más el CI. En la tabla 11.3 se muestran estos costes desglosados.

Tabla 11.3: Otros conceptos incluidos en el coste.

Desglose de costes del resto de conceptos			
Concepto	Porcentaje	Coste aplicado	Total (euros)
Gastos Generales (CI)	15 %	CD	4688.62
Beneficio Industrial	6 %	CD + CI	2156.76

El precio con IVA tanto de material como de trabajo se muestra en la tabla 11.4.

Tabla 11.4: Coste total del trabajo incluyendo el IVA.

Costes del proyecto con IVA		
Precio sin IVA (euros)	IVA (%)	Total (euros)
38102.87	21	46104.47

El coste total del proyecto con IVA supone **46104.47** euros, **CUARENTA Y SEIS MIL CIENTO CUATRO CON CUARENTA Y SIETE**.

# Bibliografía

- [1] A. A. Zhilenkov, D. D. Gilyazov, I. I. Matveev and Y. V. Krishtal, "Power line communication in IoT-systems," 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, 2017, pp. 242-245.
- [2] L. d. M. B. A. Dib, V. Fernandes, M. de L. Filomeno and M. V. Ribeiro, "Hybrid PLC/Wireless Communication for Smart Grids and Internet of Things Applications", in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 655-667, April 2018.
- [3] A. Ghiasimonfared, D. Righini, F. Marcuzzi and A. M. Tonello, "Development of a hybrid LoRa/G3-PLC IoT sensing network: An application oriented approach" *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Dresden, 2017, pp. 503-508.
- [4] J. Misurec, P. Mlynek and S. Bezzateev, "The modeling of power line for PLC in smart grids", 2017 *Progress In Electromagnetics Research Symposium - Spring (PIERS)*, St. Petersburg, 2017, pp. 780-786.
- [5] A. Sendin, J. Simon, I. Urrutia and I. Berganza, "PLC deployment and architecture for Smart Grid applications in Iberdrola", *18th IEEE International Symposium on Power Line Communications and Its Applications*, Glasgow, 2014, pp. 173-178.
- [6] L. Sadamori, S. Dominiak and T. Hunziker, "A characterization of the broadband MIMO PLC channel in aircraft", 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC), Madrid, 2017, pp. 1-6.
- [7] L. T. Berger, A. Schwager, P. Pagani and D. M. Schneider, "MIMO Power Line Communications", in *IEEE Communications Surveys & Tutorials*, vol.17, no. 1, pp.106-124, Firstquarter 2015.
- [8] Thomas Larhzaoui, Fabienne Nouvel, Jean-Yves Baudais, Pierre Degauque, Virginie Degardin. OFDM PLC transmission for aircraft flight control system. 18th IEEE International Symposium on Power Line Communications and its Applications, ISPLC 2014, 2014, Glasgow, Scotland, United Kingdom. pp.220-225, 2014, <10.1109/ISPLC.2014.6812333>. <hal-00992666>.

- [9] A. Pittolo, M. De Piante, F. Versolatto and A. M. Tonello, "In-Vehicle Power Line Communication: Differences and Similarities Among the In-Car and the In-Ship Scenarios", in *IEEE Vehicular Technology Magazine*, vol. 11, no. 2, pp. 43-51, June 2016.
- [10] F. A. Pinto- Benel and F. Cruz-Roldán, "2-ASCET for Broadband Multicarrier Transmission over In-home and In-vehicle Power Line Networks", 2015 *IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, 2015, pp. 1351-1356.
- [11] C. Cano, A. Pittolo, D. Malone, L. Lampe, A. M. Tonello and A. G. Dabak, "State of the Art in Power Line Communications: From the Applications to the Medium", in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 7, pp. 1935-1952, July 2016.
- [12] Xilinx, Inc. (2018) Zynq-7000 SoC Data Sheet: Overview.
- [13] Xilinx, Inc. (2011) AXI Reference Guide, UG761.
- [14] Xilinx, Inc. (2018) ZC706 Evaluation Board for the Zynq-7000 XC7Z045 SoC, UG954.
- [15] Ruben Nieto, Álvaro Hernández, Raúl Mateos. "Modelado HLS de un transmultiplexor Multiportadora para PLC: Comparativa de Prestaciones", Actas del Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI 2018), pp. 1-6, 2018.
- [16] P. Poudereux, Á. Hernández, F. Cruz-Roldán and R. Mateos, "FPGA-Based Architecture for Medium Access Techniques in Broadband PLC", in *IEEE Access*, vol. 6, pp. 9534-9542, 2018.
- [17] Francisco Nombela, Enrique García, Raúl Mateos, Álvaro Hernández, Real-time architecture for channel estimation and equalization in broadband PLC, *Microprocessors and Microsystems*, Volume 65, 2019, Pages 121-135.
- [18] Maxim Integrated<sup>TM</sup>, MAX2981. Integrated Powerline Communication Analog Front-End Transceiver and Line Driver, Technical Specification, 2009. <https://datasheets.maximintegrated.com/en/ds/MAX2981.pdf>
- [19] Lara Martín Lagares. Algoritmos para el Filtrado Eficiente de Señales Reales basados en la DFT. Proyecto Fin de Carrera. Universidad de Sevilla, Sevilla 2015. pp.39.
- [20] Xilinx, Inc. (2018), 7 Series DSP48E1 Slice, UG479.
- [21] IEEE Draft Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications", in *IEEE P1901/D4.01*, July 2010 , vol., no., pp.1-1589, 31 Dec. 2010.

- 
- [22] Yefersson Cañon De Antonio. Estudio de la Factibilidad Técnica de la implementación de PLC (Power Line Communication) en la red de distribución eléctrica de Bogotá. Tesis Doctoral. Universidad Nacional de Colombia, Colombia, 2016.
- [23] A. M. Tonello, F. Versolatto, B. Bejar and S. Zazo, ".<sup>A</sup> Fitting Algorithm for Random Modeling the PLC Channel", in IEEE Transactions on Power Delivery, vol. 27, no. 3, pp. 1477-1484, July 2012.
- [24] Y. H. Ma, P. L. So and E. Gunawan, "Performance analysis of OFDM systems for broadband power line communications under impulsive noise and multipath effects", in IEEE Transactions on Power Delivery, vol. 20, no. 2, pp. 674-682, April 2005.







Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá